

Evidencia de Ejercicios de Git

- Camilo Alejandro Lagos Malaver

The screenshots show the 'Aprende Git Branching' application interface. Each screenshot displays a terminal window with command history and a success message dialog box.

Screenshot 1: Level Intro

```
$ level intro
$ hint
¡Simplemente escribe 'git commit' dos veces para terminar!
$ delay 2000
$ show goal
$ objective
$ git commit
$ git commit
```

¡Buen trabajo!!!
Resolviste el nivel en 2 comandos; nuestra mejor solución usa 2.
¡Fabuloso! Igualaste o superaste nuestra solución.
¿Quieres pasar al nivel "Creando ramas en Git", el próximo nivel?

Screenshot 2: Level Creando ramas en Git

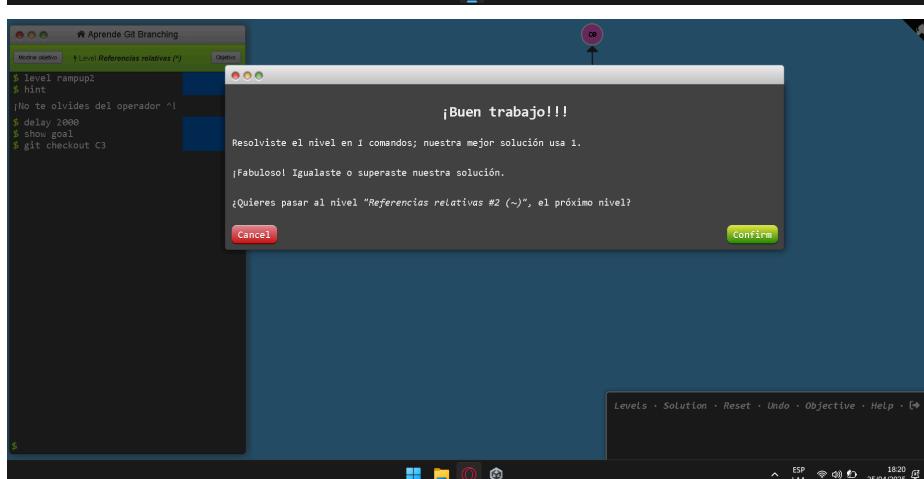
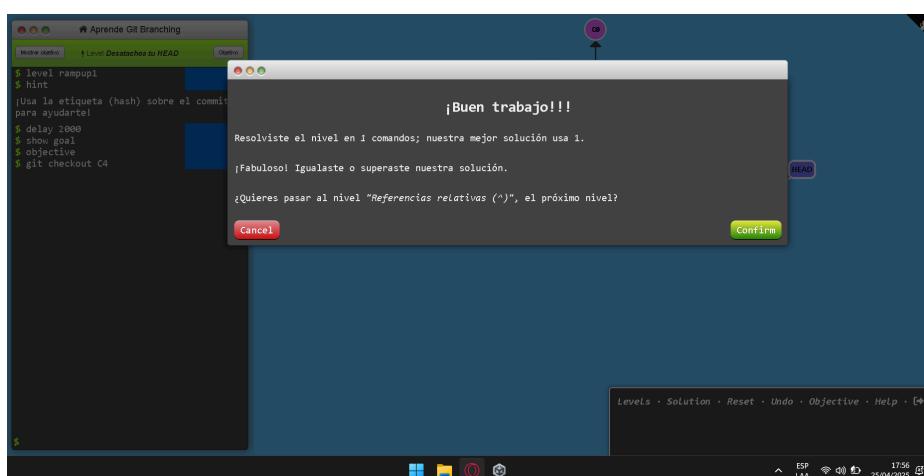
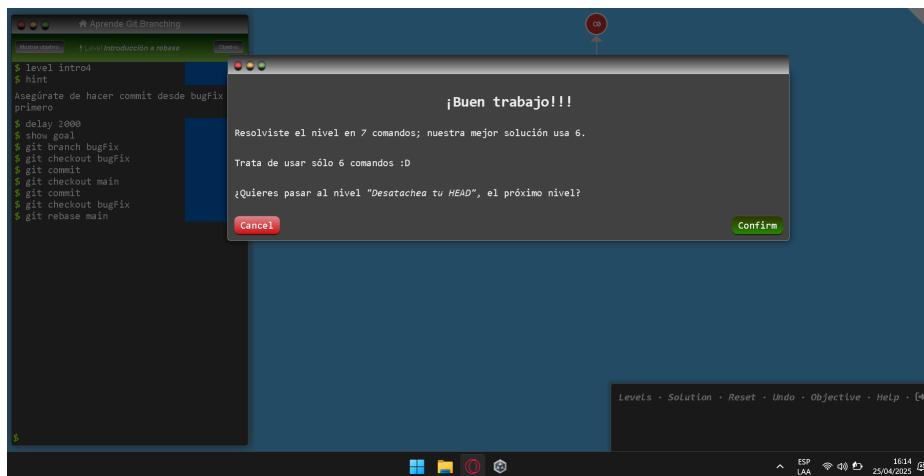
```
$ level intro2
$ hint
Crea una nueva rama con "git branch [nombre]" y sitúate en ella con "git checkout [nombre]"
$ delay 2000
$ show goal
$ git branch bugFix
$ git checkout bugFix
```

¡Buen trabajo!!!
Resolviste el nivel en 2 comandos; nuestra mejor solución usa 2.
¡Fabuloso! Igualaste o superaste nuestra solución.
¿Quieres pasar al nivel "Haciendo merge en Git", el próximo nivel?

Screenshot 3: Level Haciendo merge en Git

```
$ level intro3
$ hint
Acuérdate de hacer commit en el orden especificado (bugFix antes de main)
$ delay 2000
$ show goal
$ objective
$ git branch bugFix
El comando "git branch bugFix" no está soportado, ¡disculpa!
$ git branch bugFix
$ git checkout bugFix
$ git merge main
$ git commit
$ git merge bugFix
```

¡Buen trabajo!!!
Resolviste el nivel en 6 comandos; nuestra mejor solución usa 5.
Trata de usar sólo 5 comandos :D
El comando "git branch bugFix" no está soportado, ¡disculpa!
¿Quieres pasar al nivel "Introducción a rebase", el próximo nivel?



¡Buen trabajo!!!

Resolviste el nivel en 14 comandos; nuestra mejor solución usa 3.

Trata de usar sólo 3 comandos :D

¿Quieres pasar al nivel "Revirtiendo cambios en git", el próximo nivel?

Confirm

Level References relatives #2

```
$ level rampup3
$ hint
Vas a necesitar usar al menos una referencia directa (hash) para completar este nivel
$ delay 2000
$ show goal
$ git branch -f bugFix HEAD~2
$ git branch -f main HEAD~1
$ git checkout main
$ git commit
$ git checkout C1
$ git checkout C1
$ git checkout C6
$ git branch -f main HEAD~1
$ git commit
(Cuidado!) Modo de HEAD separado (detached HEAD)
$ git checkout C5
$ git checkout C6
$ git checkout C8
$ git branch -f main HEAD~1
$ git checkout C1
```

Levels · Solution · Reset · Undo · Objective · Help · ⌂

Level Revertiendo cambios en git

separado. Usa checkout si quieres moverte.

```
$ git checkout local
$ git reset head
La referencia head no existe o es desconocida
$ git reset HEAD
$ git revert HEAD
$ undo
$ git reset HEAD
$ git reset local
$ git checkout C1
$ git reset HEAD
No puedes hacer reset en el modo separado. Usa checkout si quieres moverte.
$ git checkout main
$ git reset HEAD
$ git reset main
$ git checkout C3
$ git checkout C3'
$ git checkout HEAD
$ git checkout pushed
$ git checkout C2'
$ git branch -f pushed HEAD
$ git checkout C3
$ git checkout C1
$ git branch -f local HEAD
$
```

¡Buen trabajo!!!

Resolviste el nivel en 20 comandos; nuestra mejor solución usa 3.

Trata de usar sólo 3 comandos :D

¿Quieres pasar al nivel "Introducción a cherry-pick", el próximo nivel?

Confirm

Level Revertiendo cambios en git

separado. Usa checkout si quieres moverte.

```
$ git checkout local
$ git reset head
La referencia head no existe o es desconocida
$ git reset HEAD
$ git revert HEAD
$ undo
$ git reset HEAD
$ git reset local
$ git checkout C1
$ git reset HEAD
No puedes hacer reset en el modo separado. Usa checkout si quieres moverte.
$ git checkout main
$ git reset HEAD
$ git reset main
$ git checkout C3
$ git checkout C3'
$ git checkout HEAD
$ git checkout pushed
$ git checkout C2'
$ git branch -f pushed HEAD
$ git checkout C3
$ git checkout C1
$ git branch -f local HEAD
$
```

¡Buen trabajo!!!

Resolviste el nivel en 8 comandos; nuestra mejor solución usa 1.

Trata de usar sólo 1 comandos :D

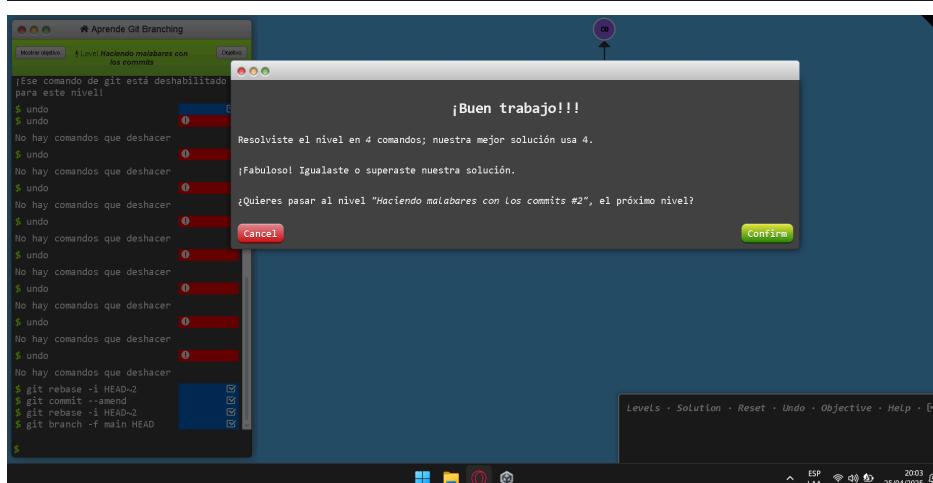
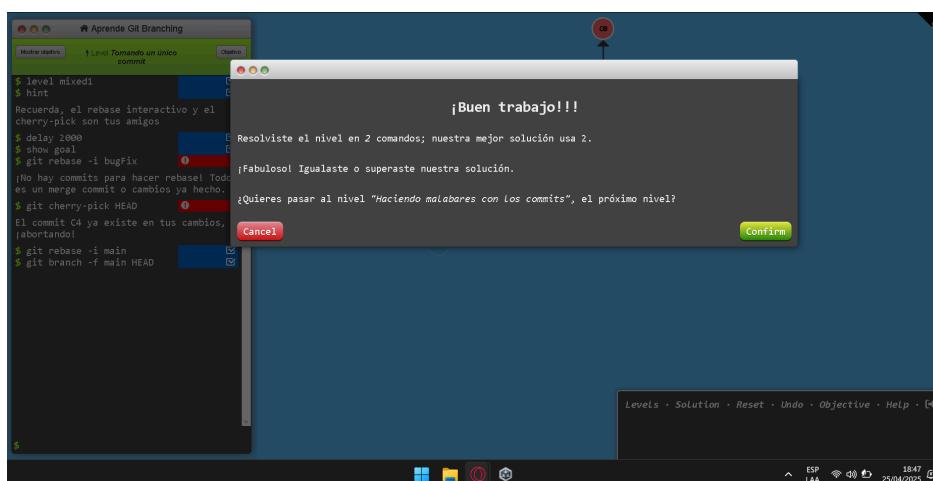
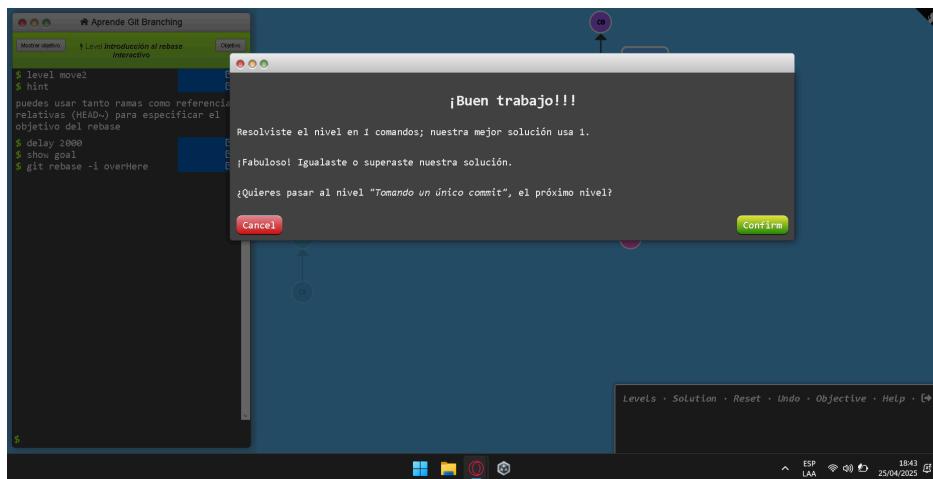
¿Quieres pasar al nivel "Introducción al rebase interactivo", el próximo nivel?

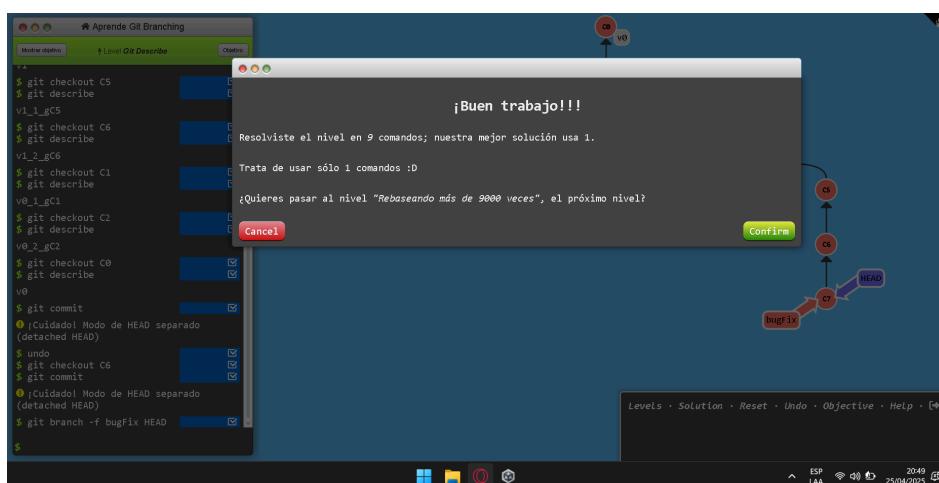
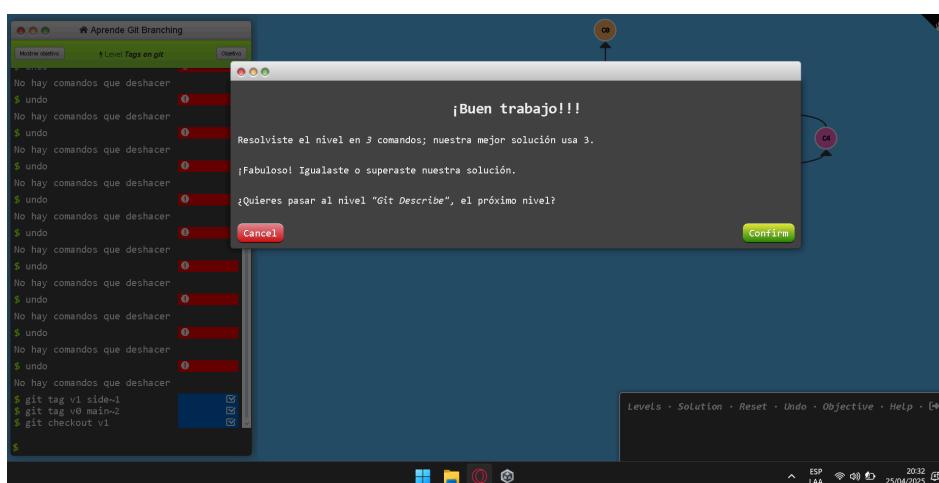
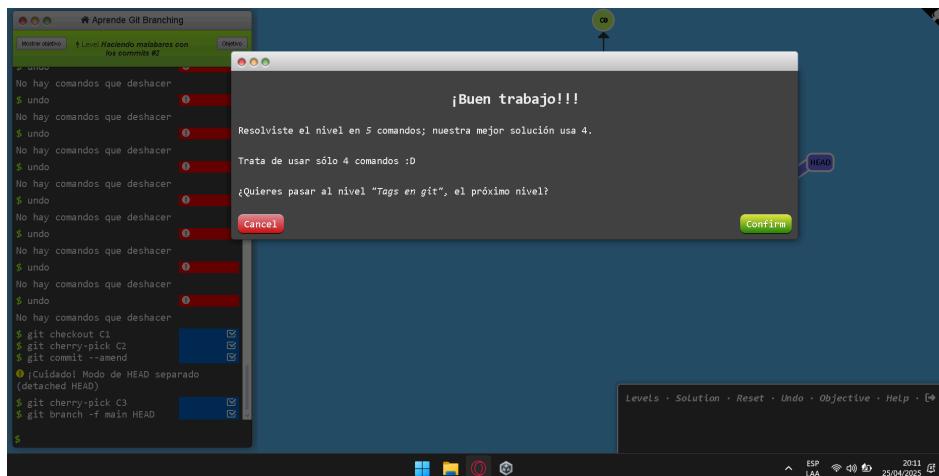
Confirm

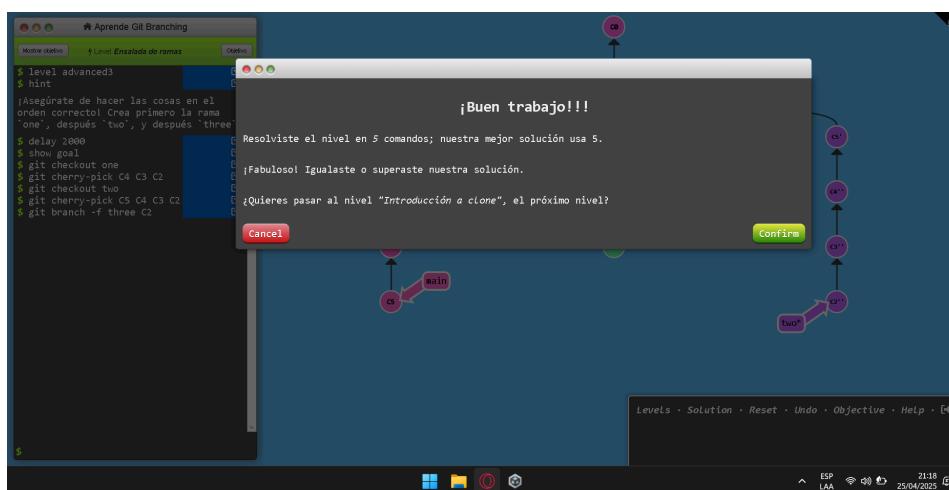
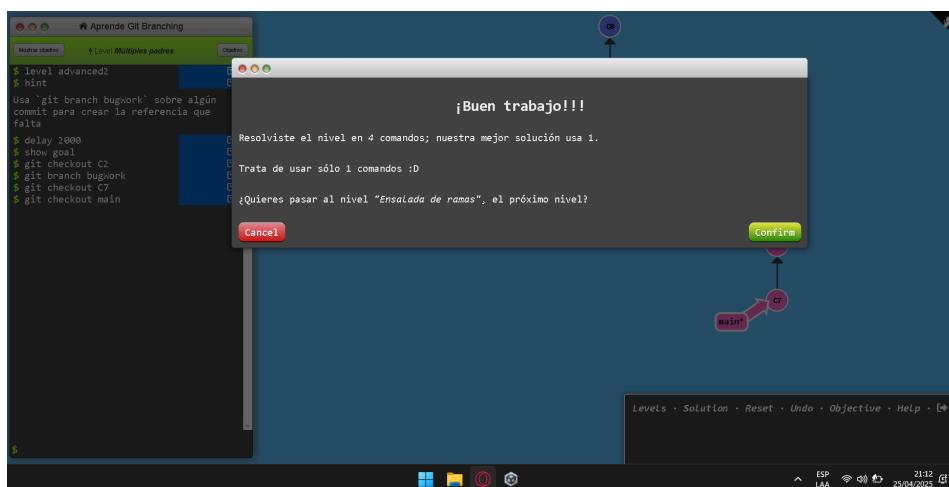
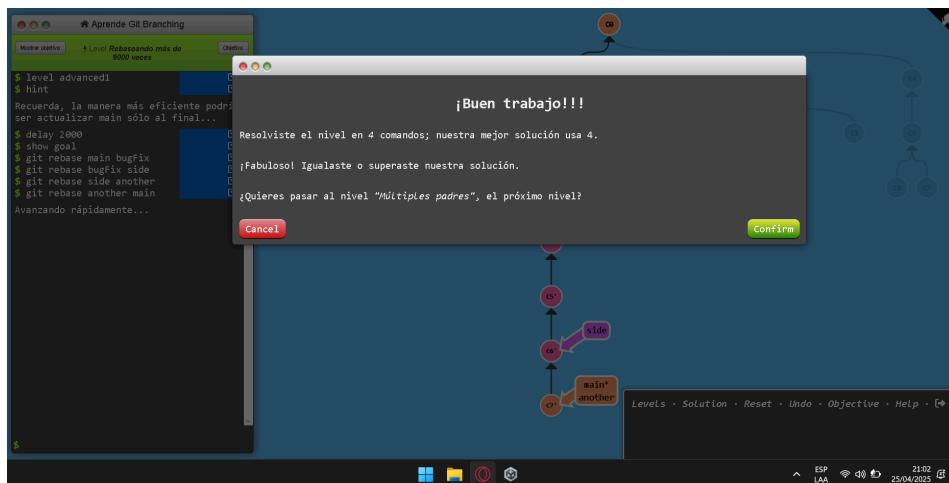
Level Introducción a cherry-pick

```
$ level move1
$ hint
git cherry-pick seguido de los nombres de los commits
$ delay 2000
$ show goal
$ git revert bugFix
$ git checkout C5
$ git branch -f side HEAD~1
$ git revert side
$ git cherry-pick HEAD
El commit C4' ya existe en tus cambios. [abortar]
$ undo
$ undo
$ git branch -f side HEAD~1
$ git checkout C4
$ git revert side
$ undo
$ git revert HEAD
$ undo
$ git checkout C3'
$ git revert side
$ git revert another
$ git branch -f main HEAD
```

Levels · Solution · Reset · Undo · Objective · Help · ⌂







Aprende Git Branching

Mostrar código | Nivel Introducción a clone | Objetivo

```
$ level remotel
$ hint
¡Simplemente escribe 'git clone'!
$ delay 2000
$ show goal
$ git clone
local branch "main" set to track
remote branch "origin/main"
```

Buen trabajo!!!

Resolviste el nivel en 1 comandos; nuestra mejor solución usa 1.

¡Fabuloso! Igualaste o superaste nuestra solución.

¿Quieres pasar al nivel "Ramas remotas", el próximo nivel?

Cancel Confirm

Levels · Solution · Reset · Undo · Objective · Help · ↗

ESP LAA 21:29 25/04/2025

Aprende Git Branching

Mostrar código | Nivel Rama remotas | Objetivo

```
$ level remote2
$ hint
Presta atención al orden: ¡has commit sobre main primero!
$ delay 2000
$ show goal
$ git commit
$ git checkout origin/main
$ git commit
$ git commit
Cuidado! Modo de HEAD separado (detached HEAD)
```

Buen trabajo!!!

Resolviste el nivel en 3 comandos; nuestra mejor solución usa 3.

¡Fabuloso! Igualaste o superaste nuestra solución.

¿Quieres pasar al nivel "git fetch", el próximo nivel?

Cancel Confirm

Levels · Solution · Reset · Undo · Objective · Help · ↗

ESP LAA 21:41 25/04/2025

Aprende Git Branching

Mostrar código | Nivel git fetch | Objetivo

```
$ level remote3
$ hint
Simplemente escribe git fetch!
$ delay 2000
$ show goal
$ git fetch
```

Buen trabajo!!!

Resolviste el nivel en 1 comandos; nuestra mejor solución usa 1.

¡Fabuloso! Igualaste o superaste nuestra solución.

¿Quieres pasar al nivel "git pull", el próximo nivel?

Cancel Confirm

Levels · Solution · Reset · Undo · Objective · Help · ↗

ESP LAA 21:50 25/04/2025

¡Buen trabajo!!!

Resolviste el nivel en 1 comandos; nuestra mejor solución usa 1.

¡Fabulos! Igualaste o superaste nuestra solución.

¿Quieres pasar al nivel "Simulando el trabajo en equipo", el próximo nivel?

Cancel Confirm

```
$ Aprende Git Branching
$ level remote4
$ hint
Simplemente ejecuta git pull!
$ delay 2000
$ show goal
$ git pull
```

Levels · Solution · Reset · Undo · Objective · Help · [?

ESP LAA 21:51 25/04/2025

¡Buen trabajo!!!

Resolviste el nivel en 4 comandos; nuestra mejor solución usa 4.

¡Fabulos! Igualaste o superaste nuestra solución.

¿Quieres pasar al nivel "git push", el próximo nivel?

Cancel Confirm

```
$ Aprende Git Branching
$ level remote5
$ hint
Recuerda que puedes especificar cuántos commits simular
$ delay 2000
$ show goal
$ git clone
$ local branch "main" set to track
remote branch "o/main"
$ git fakeTeamwork foo 2
$ git commit
$ git push
$ git pull
```

Levels · Solution · Reset · Undo · Objective · Help · [?

ESP LAA 21:54 25/04/2025

¡Buen trabajo!!!

Resolviste el nivel en 3 comandos; nuestra mejor solución usa 3.

¡Fabulos! Igualaste o superaste nuestra solución.

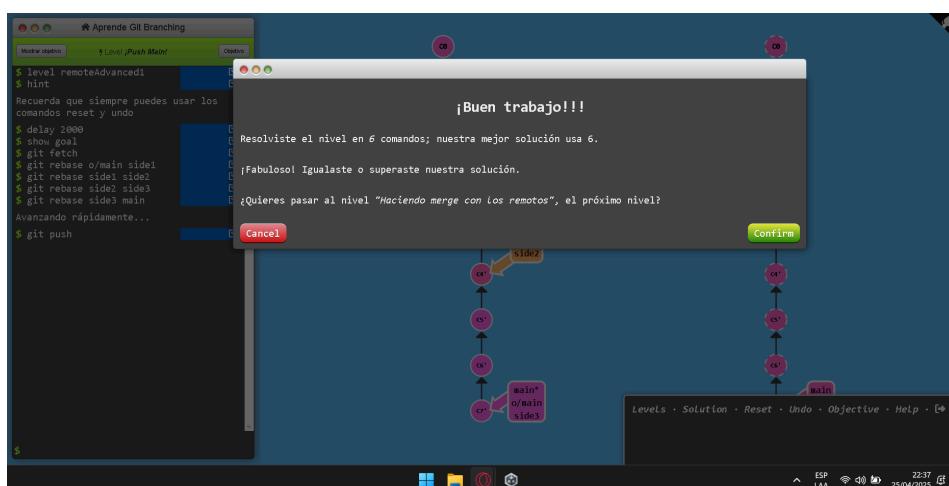
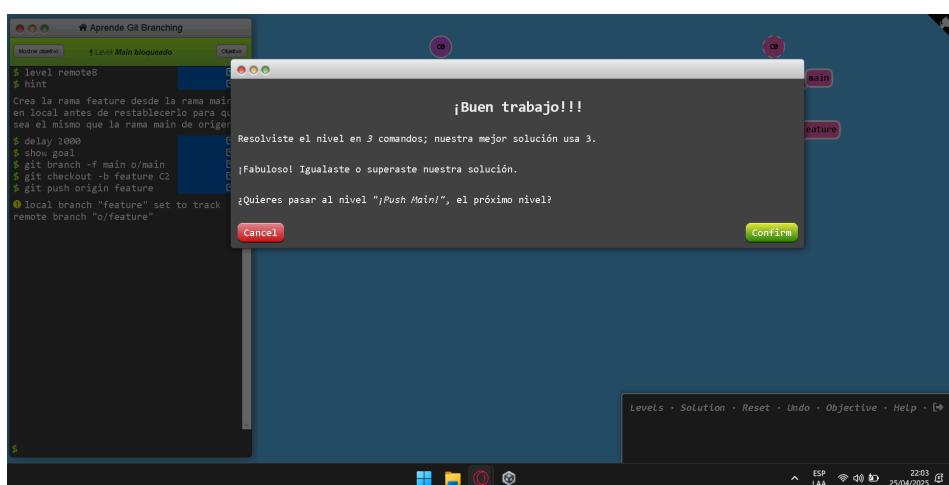
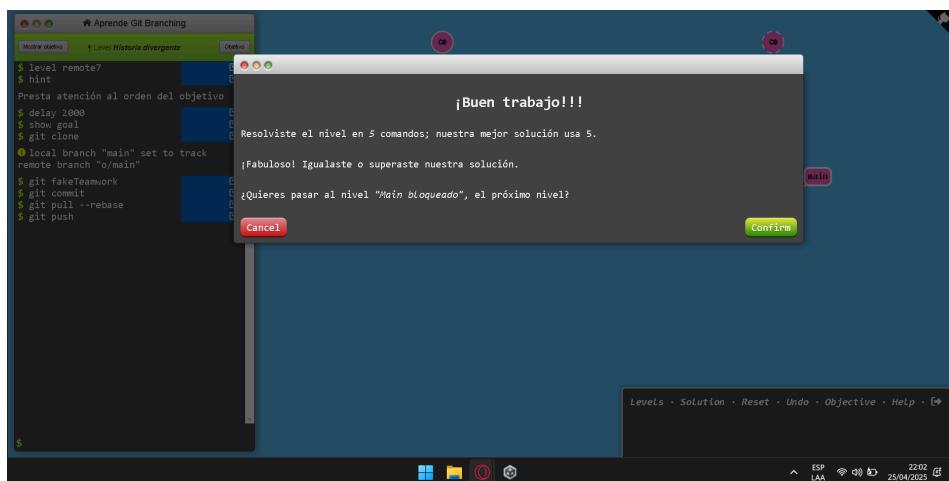
¿Quieres pasar al nivel "Historia divergente", el próximo nivel?

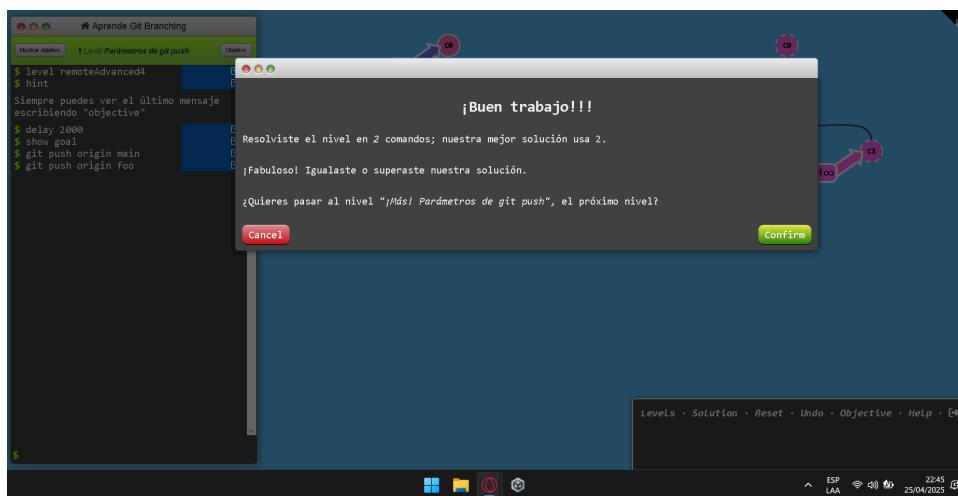
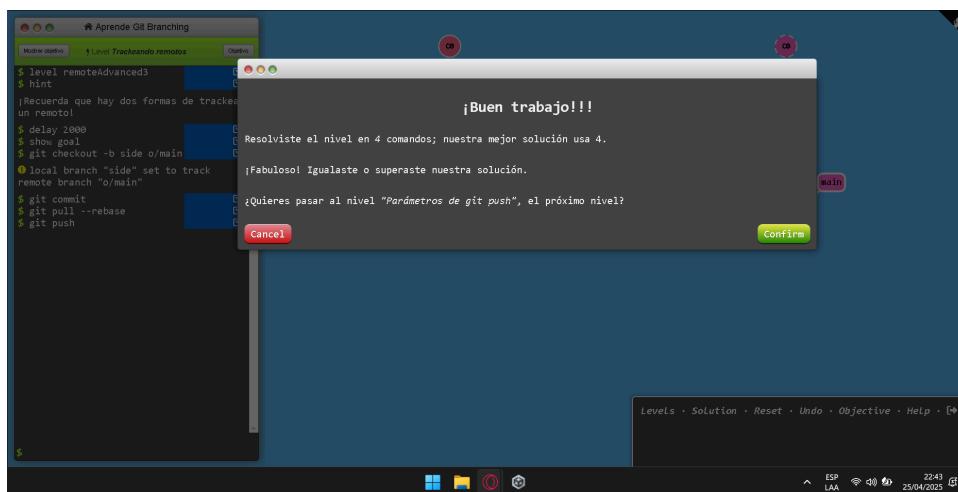
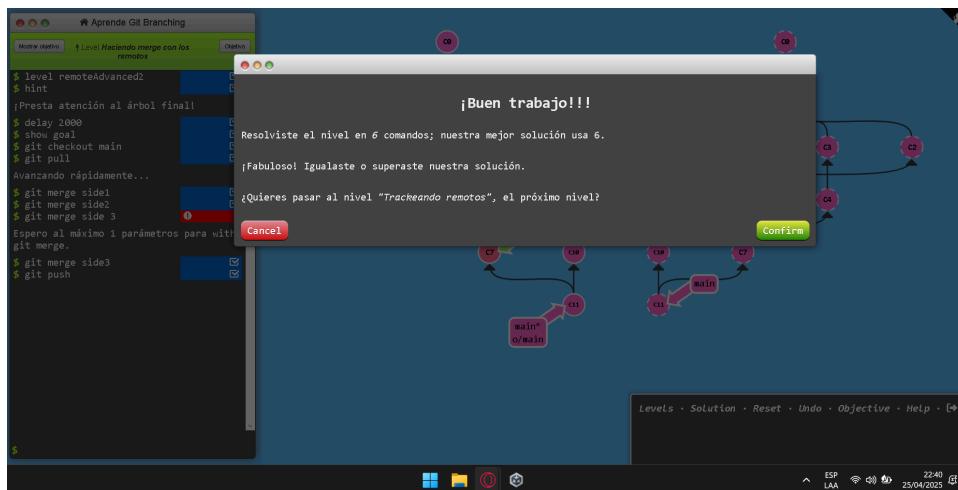
Cancel Confirm

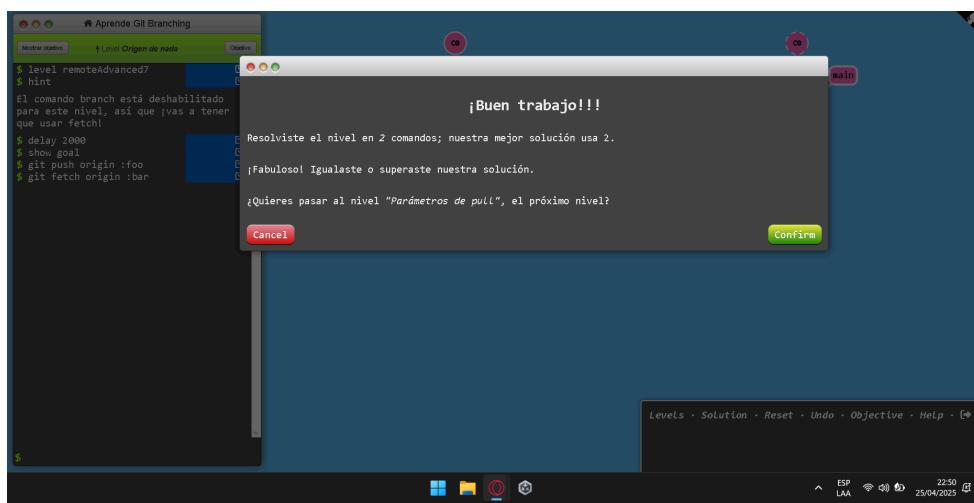
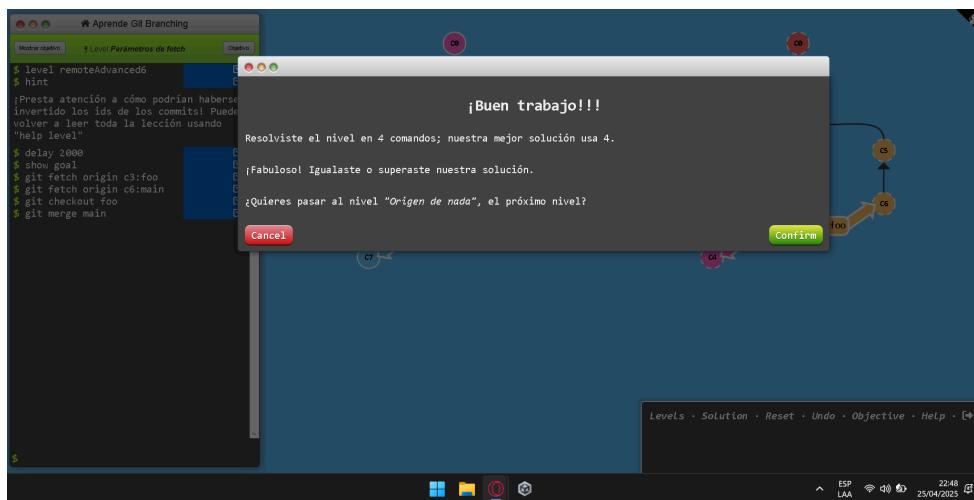
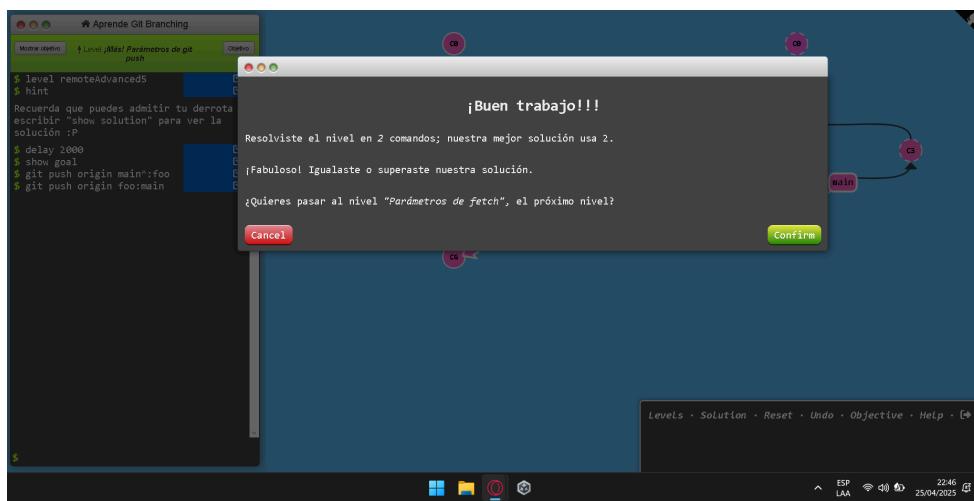
```
$ Aprende Git Branching
$ level remote6
$ hint
¡Recuerda que tienes que clonar antes de hacer push!
$ delay 2000
$ show goal
$ git clone
$ git commit
$ git branch -f o/main main
No puedes ejecutar ese comando en una rama remota
$ git push
```

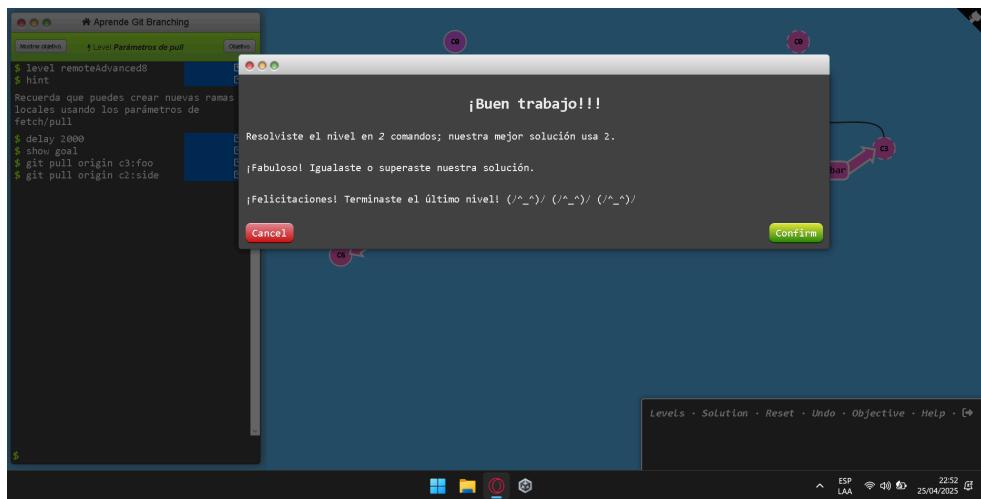
Levels · Solution · Reset · Undo · Objective · Help · [?

ESP LAA 21:57 25/04/2025









- Sergio Nicolás Correa Escobar

The screenshot shows a web-based application for learning Git branching. On the left, a terminal window displays the command:

```
$ git checkout v1
```

On the right, there are two main components: a diagram illustrating commit history and a detailed explanation of merge parents.

Diagram (Top Right):

The diagram shows a commit graph with several commits (C0, C1, C2, C3, C4, v0, v1, v2, v3, v4) and tags (main, side). Commit C0 is at the top. Commit C1 is above C0. Commit C2 is to the left of C1. Commit C3 is below C2. Commit C4 is to the right of C1. Tag v0 is associated with C1. Tag v1 is associated with C2. Tag v2 is associated with C3. Tag v3 is associated with C4. Tag v4 is associated with C1. A pink arrow labeled "main" points from C1 to C4. A green arrow labeled "side" points from C2 to C3. A blue arrow labeled "HEAD" points from C1 to C2.

Explanation Window (Bottom Right):

The window title is "Objetivo a cumplir". It contains the text:

Puedes ocultar esta ventana con "Hide goal."

The diagram in this window is identical to the one above it, showing the same commit graph and tag associations.

Terminal Window (Bottom Left):

The title bar says "Aprende Git Branching" and "Mostrar objetivo". The terminal shows the command:

```
$ level advanced2
```

Below the command, there is explanatory text about merge parents:

Especificando los padres

Como el modificador de `~`, `^` también acepta un número opcional después de él.

En lugar de especificar cuántas generaciones hacia atrás ir (como `~`), el modificador de `^` especifica por cuál de las referencias padres seguir en un commit de merge. Recuerda que un commit de merge tiene múltiples padres, por lo que el camino a seguir es ambiguo.

Git normalmente sigue el "primer" parent de un commit de merge, pero especificando un número junto con `^` cambia este comportamiento predefinido.

Demasiada charla, veámoslo en acción.

At the bottom of the window are red and green navigation arrows.

learngitbranching.js.org/?locale=es_ES&NODEMO=

Aprende Git Branching

Mostrar objetivo | **Level Ensalada de ramas** | Objetivo

```
$ level advanced3
$ hint
¡Asegúrate de hacer las cosas en orden correcto! Crea primero 'one', después 'two', y después 'delay 2000'.
$ delay 2000
$ show goal
$ git rebase -i one
$ reset
$ git checkout one
El comando "git checkout one" no existe, ¡disculpa!
$ git cherry-pick C4 C3 C2
El commit C4 ya existe en tus cambios, abortando!
$ reset
$ git checkout one
$ git cherry-pick C4 C3 C2
$ git checkout two
$ git cherry-pick C5 C4 C3 C2
$ git rebase C2 three
Avanzando rápidamente...
```

¡Buen trabajo!!!

Resolviste el nivel en 5 comandos; nuestra mejor solución usa 5.

¡Fabuloso! Igualaste o superaste nuestra solución.

¿Quieres pasar al nivel "Introducción a clone", el próximo nivel?

Confirm

Levels · Solution · Reset · Undo · Objective · Help · [✖]

learngitbranching.js.org/?locale=es_ES&NODEMO=

Aprende Git Branching

Ocultar objetivo | **Level Introducción a cloning** | Cumplir

1: Desatachea tu HEAD

Moviendo el trabajo por ahí

Ponte cómodo cuando modifiques el directorio fuente

1: Introducción a cherry-pick

Un poco de todo

Un batiburrillo de técnicas, trucos y sugerencias sobre Git

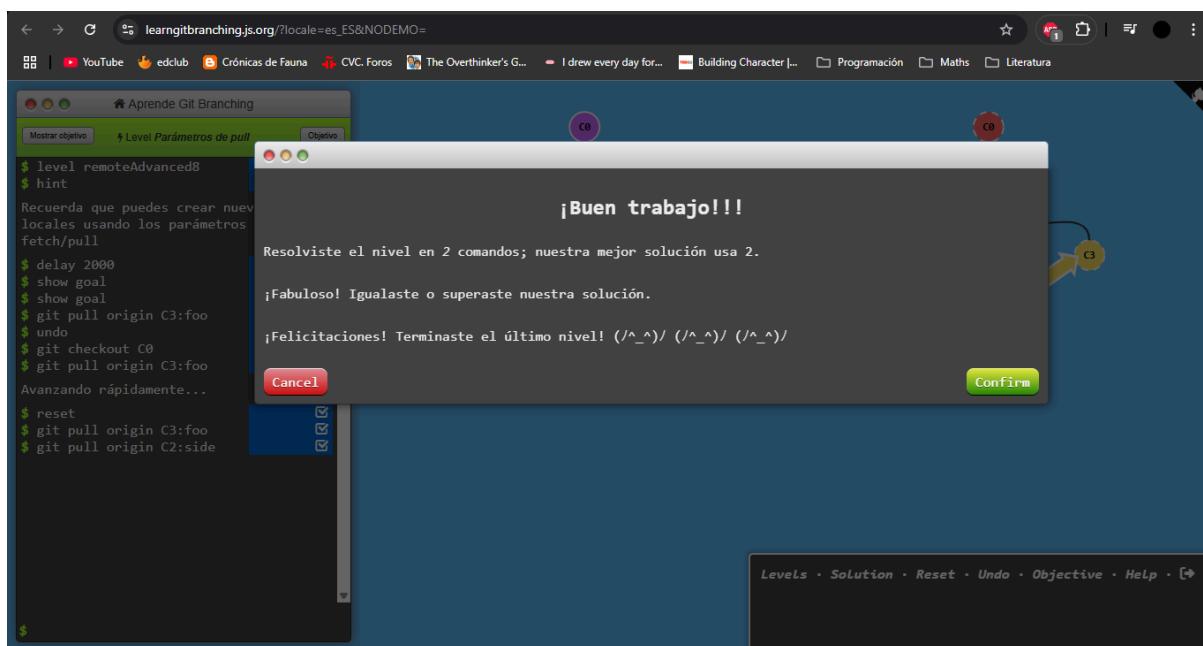
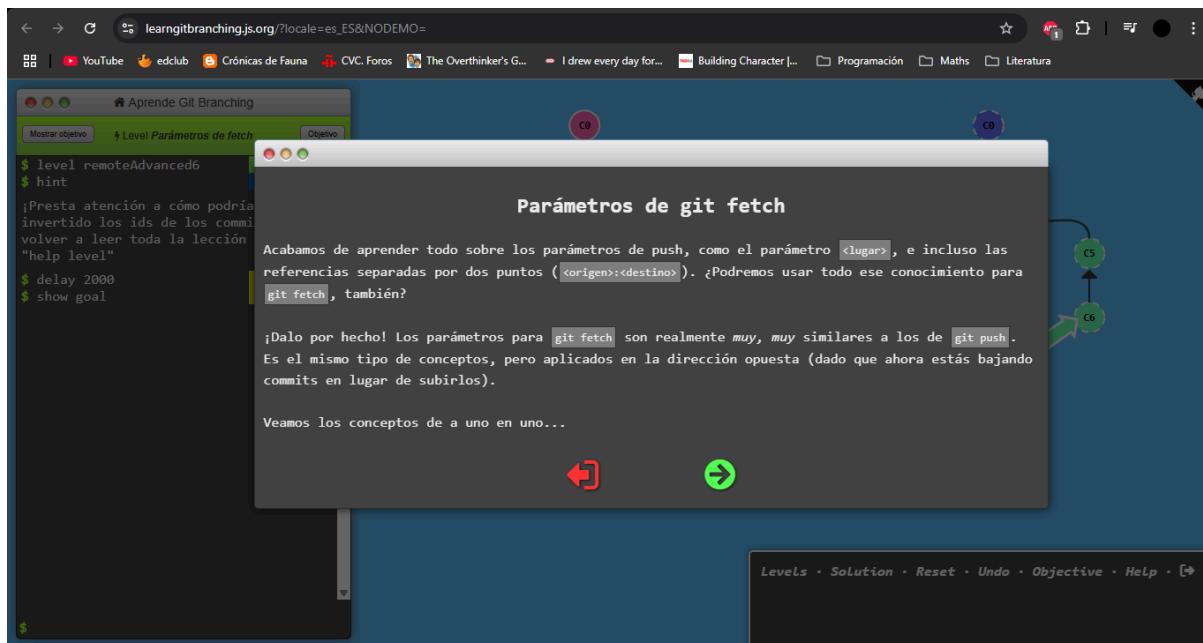
1: Tomando un único commit

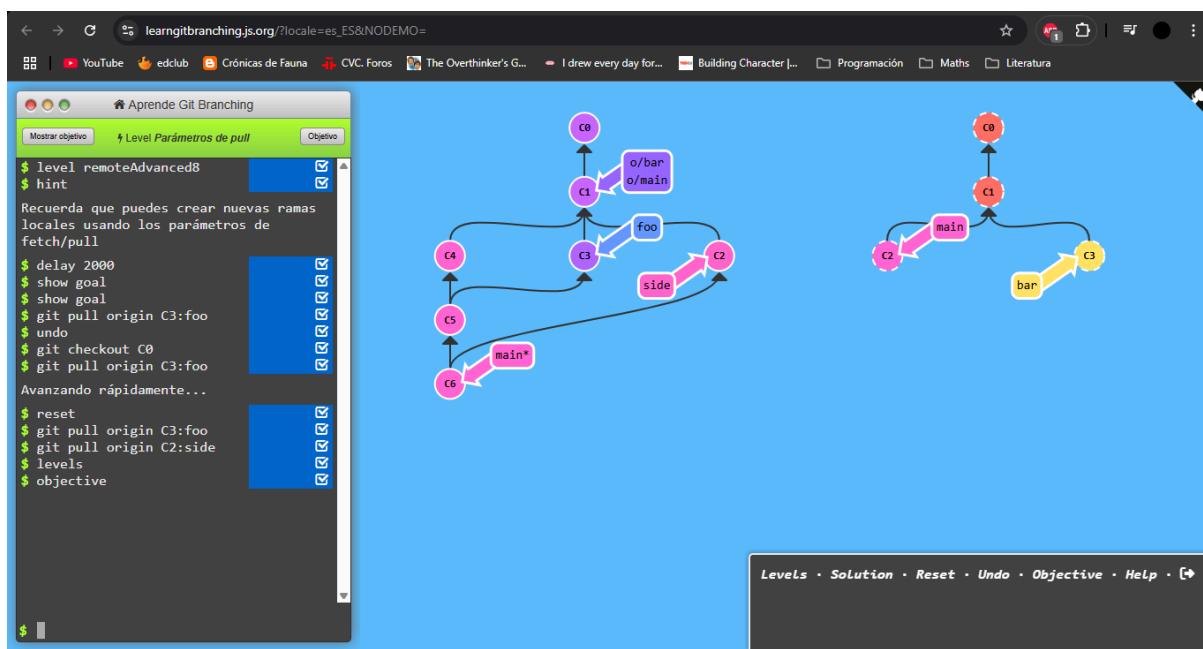
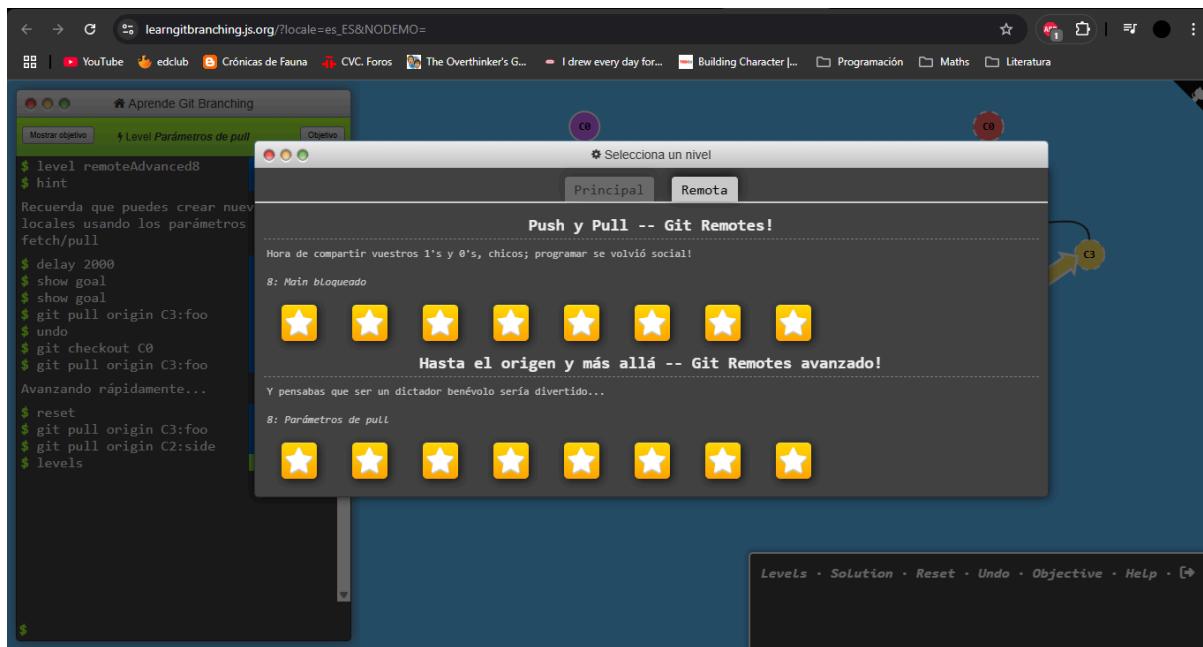
Temas avanzados

¡Para los verdaderos valientes!

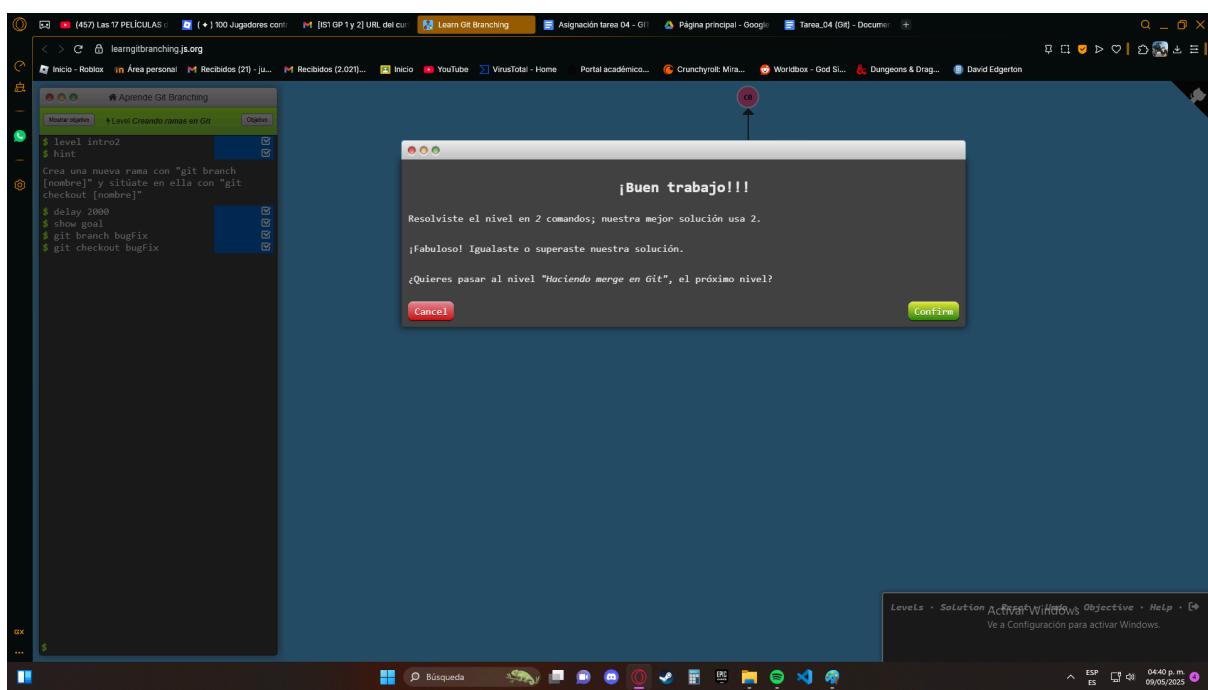
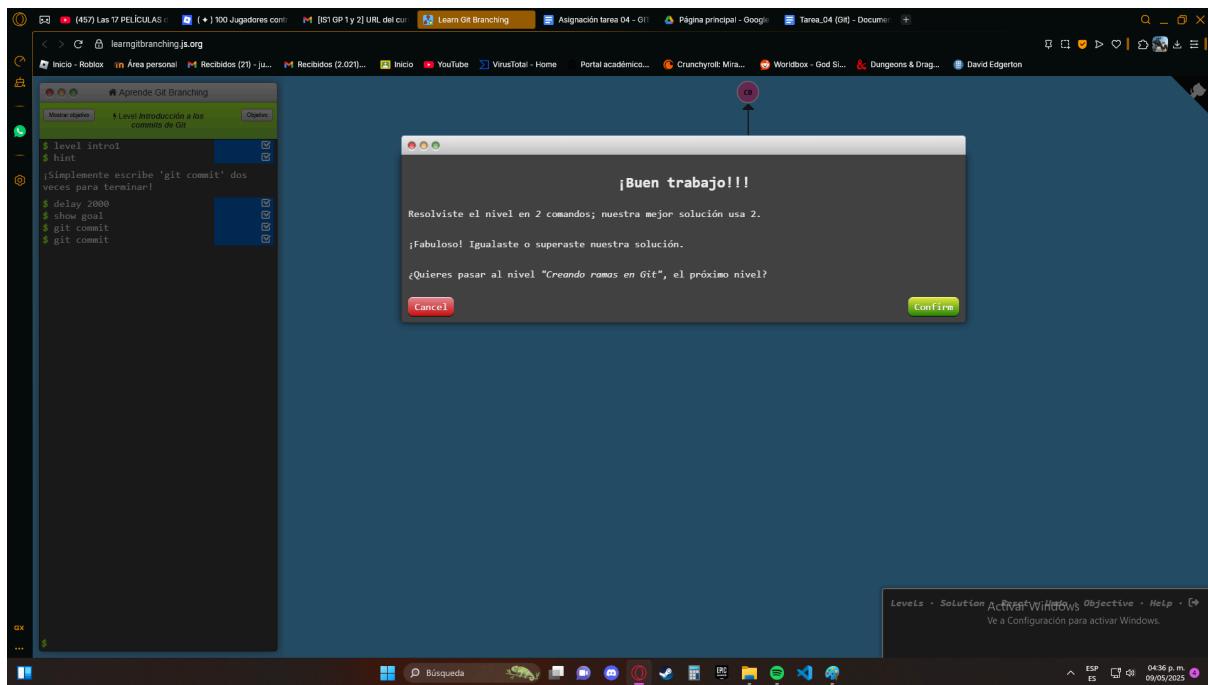
3: Ensalada de ramas

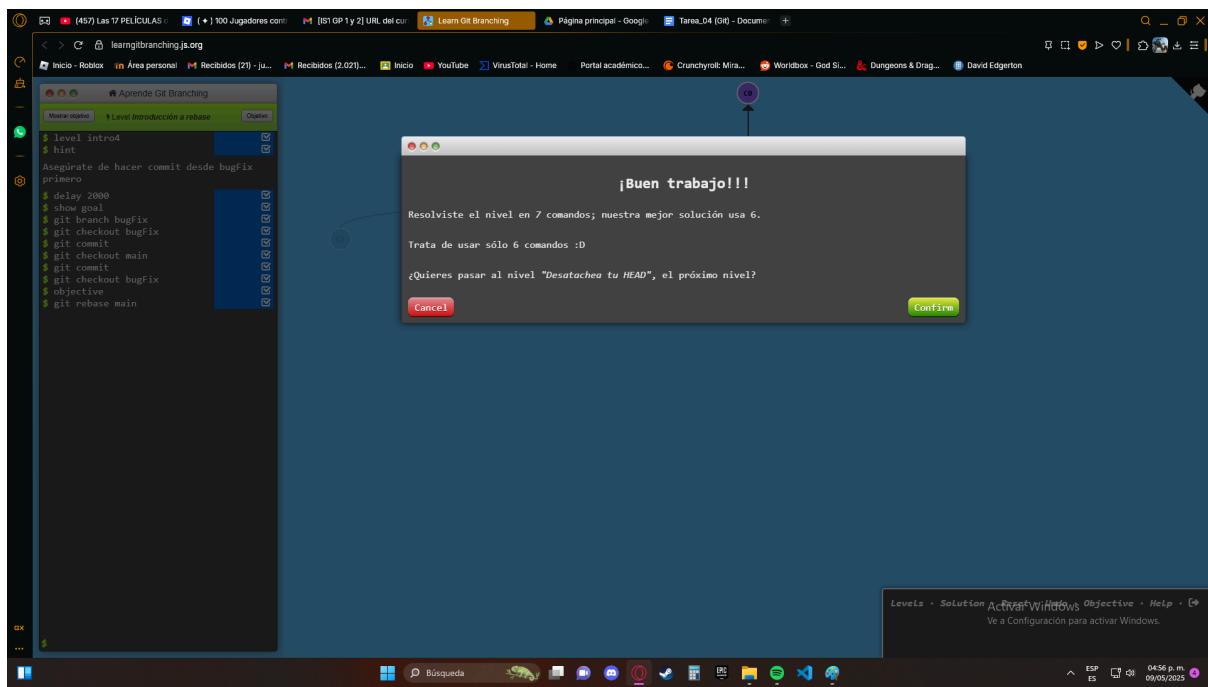
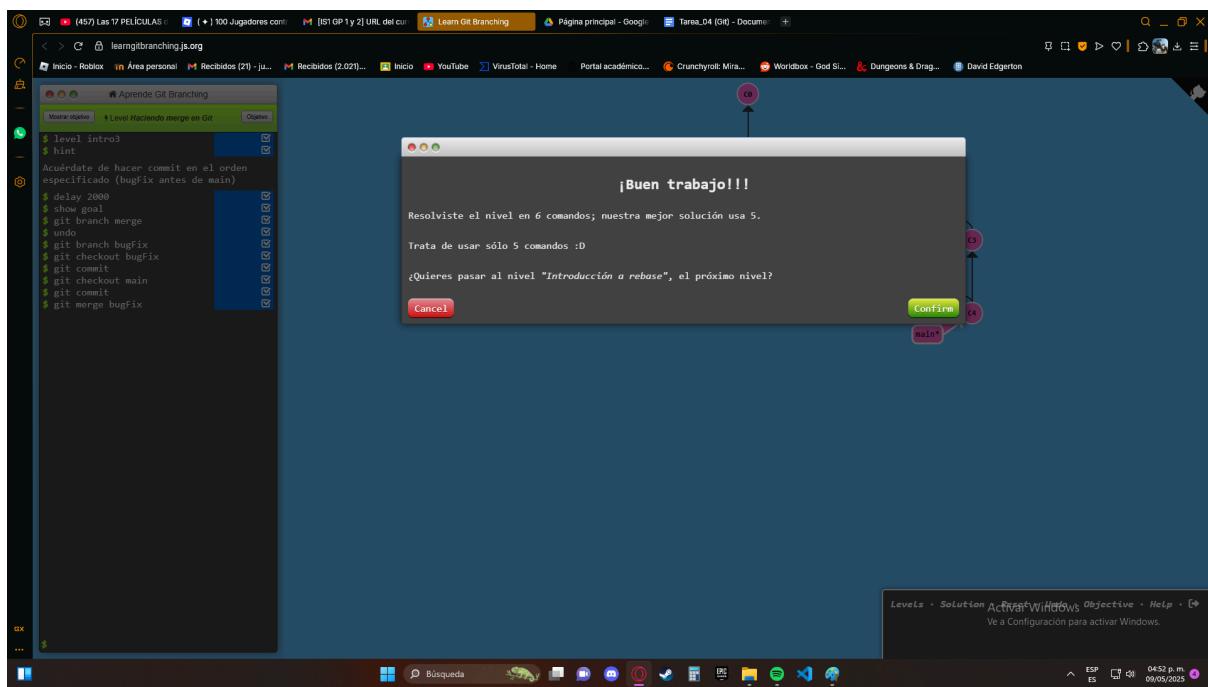
Objective · Help · [✖]

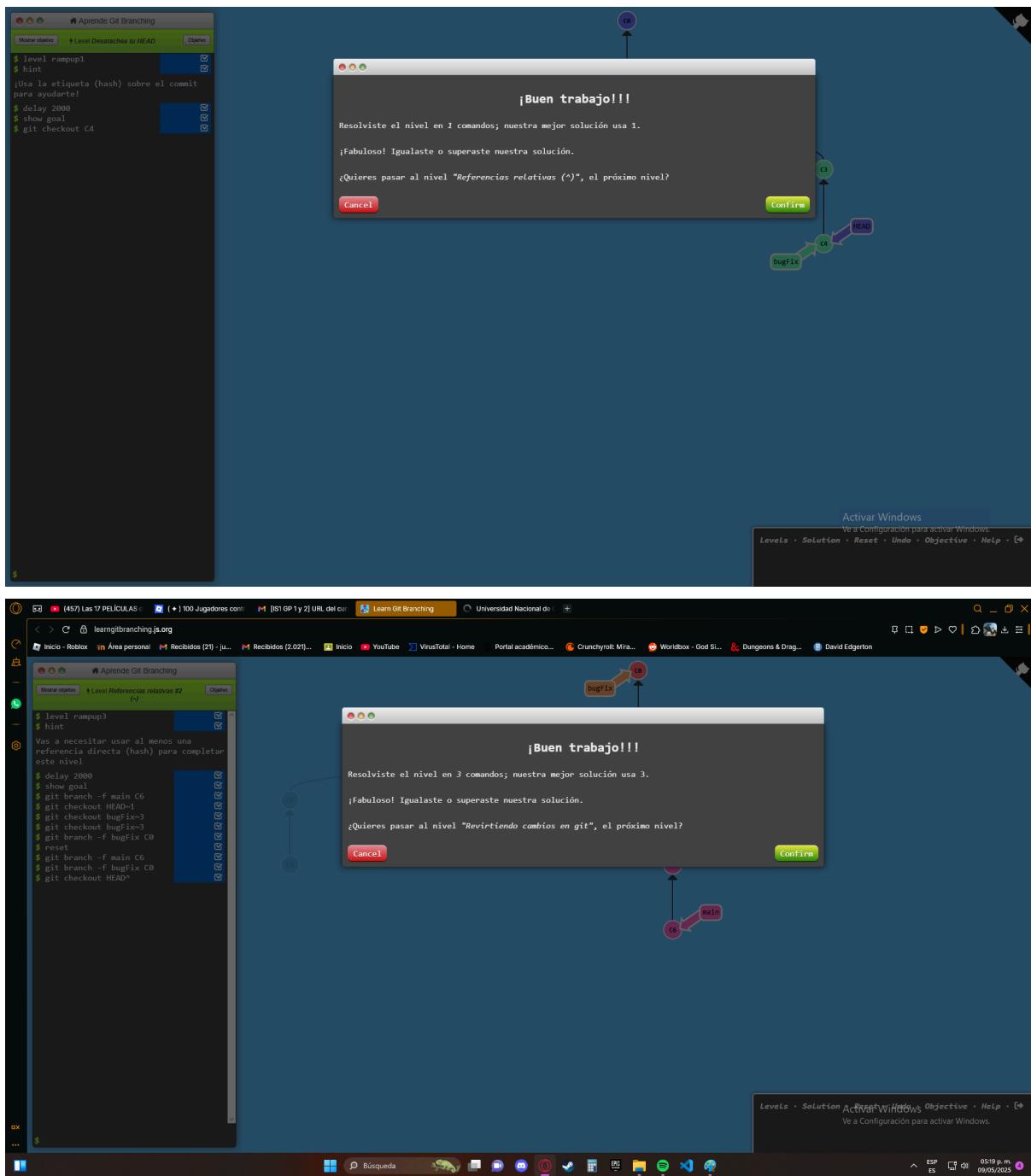


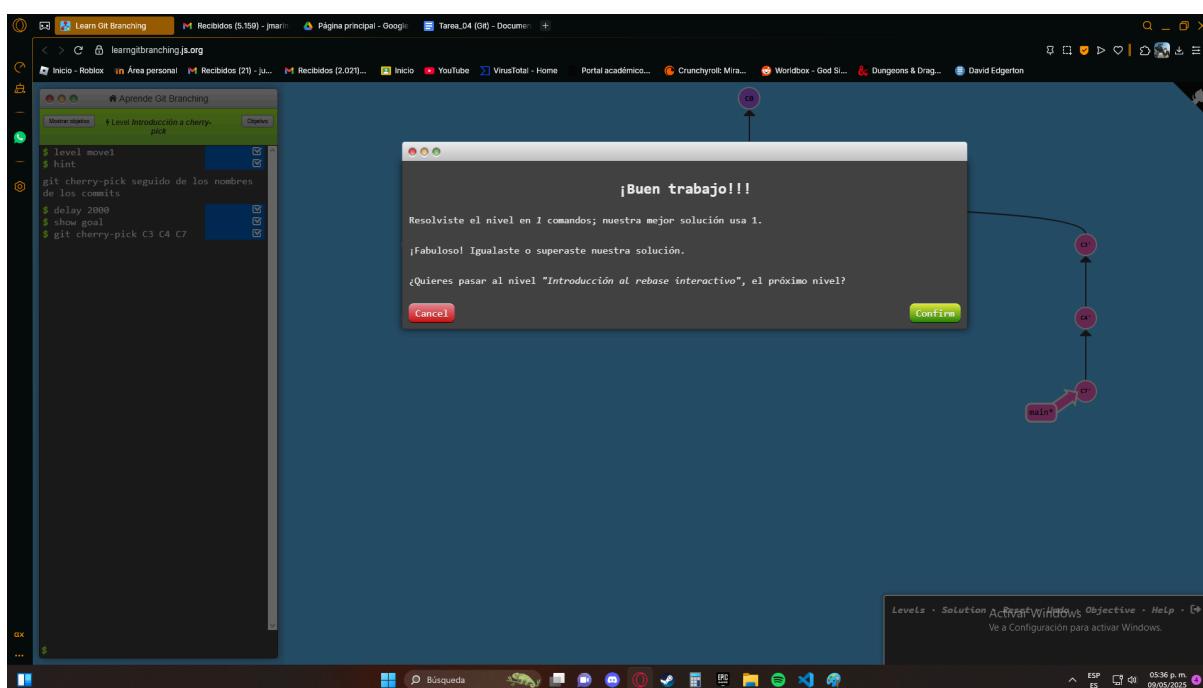
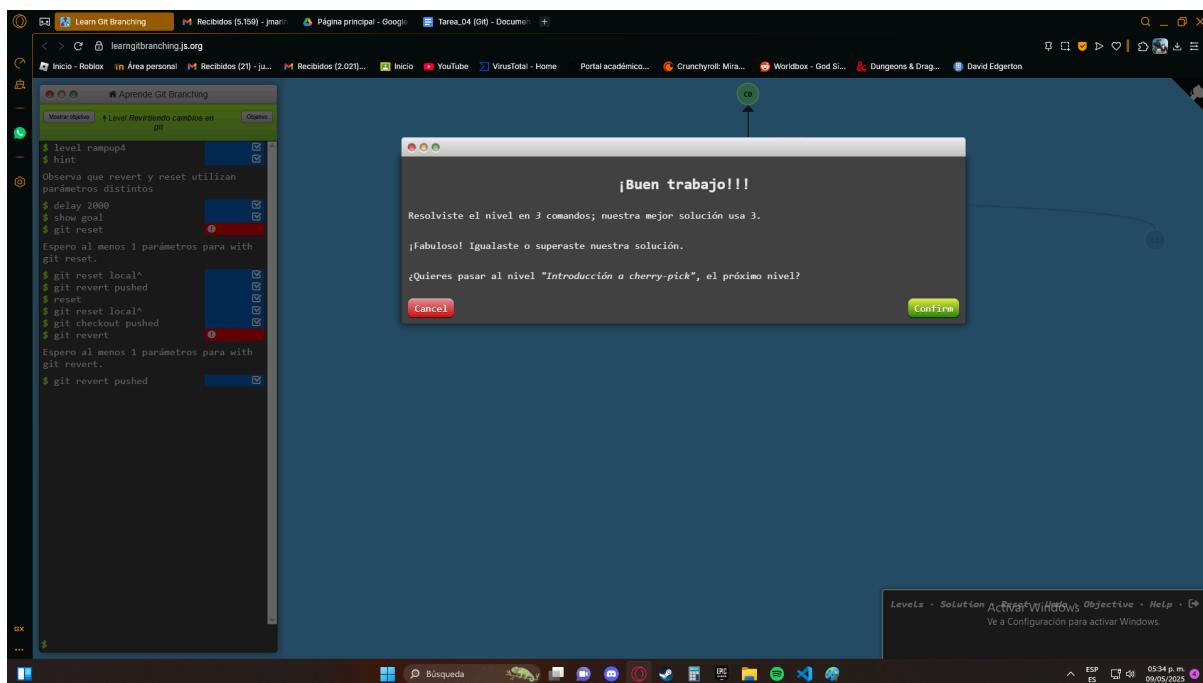


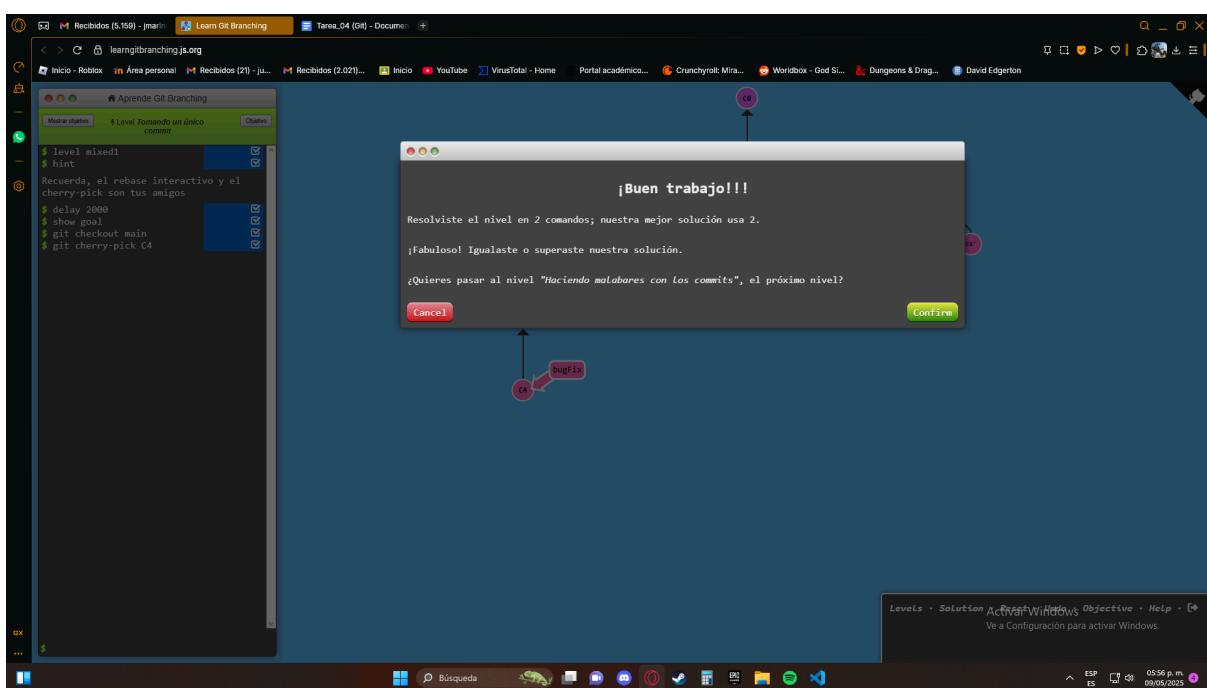
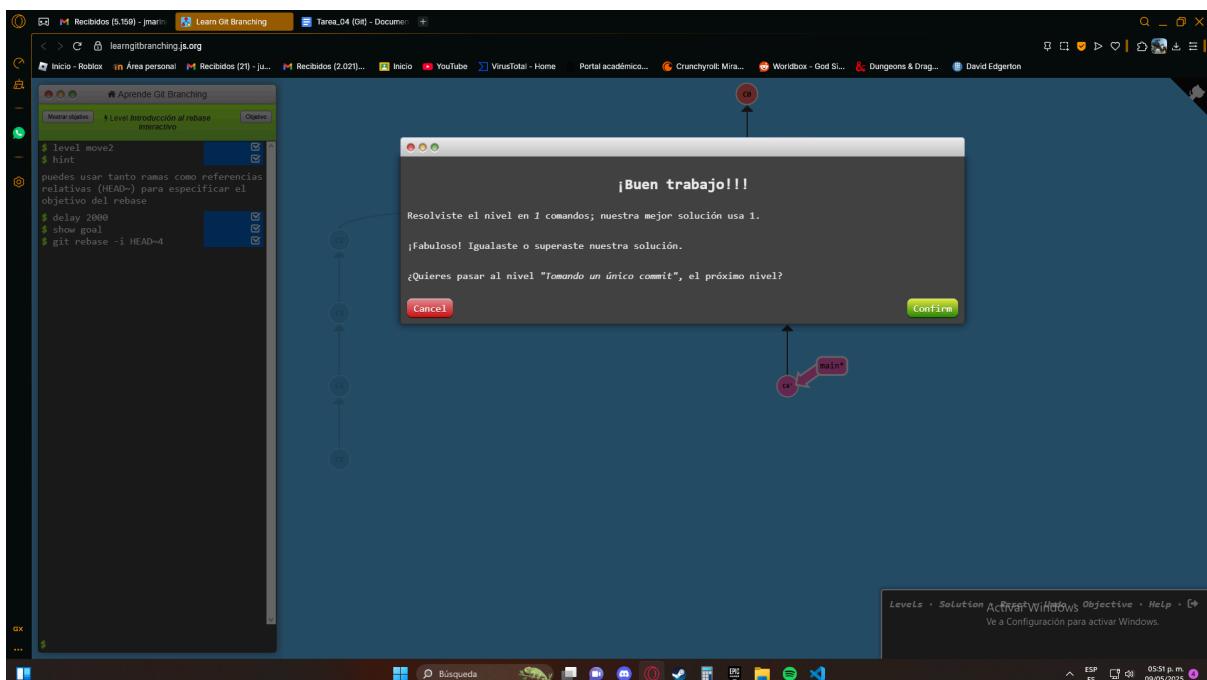
- Juan Felipe Marín Moreno

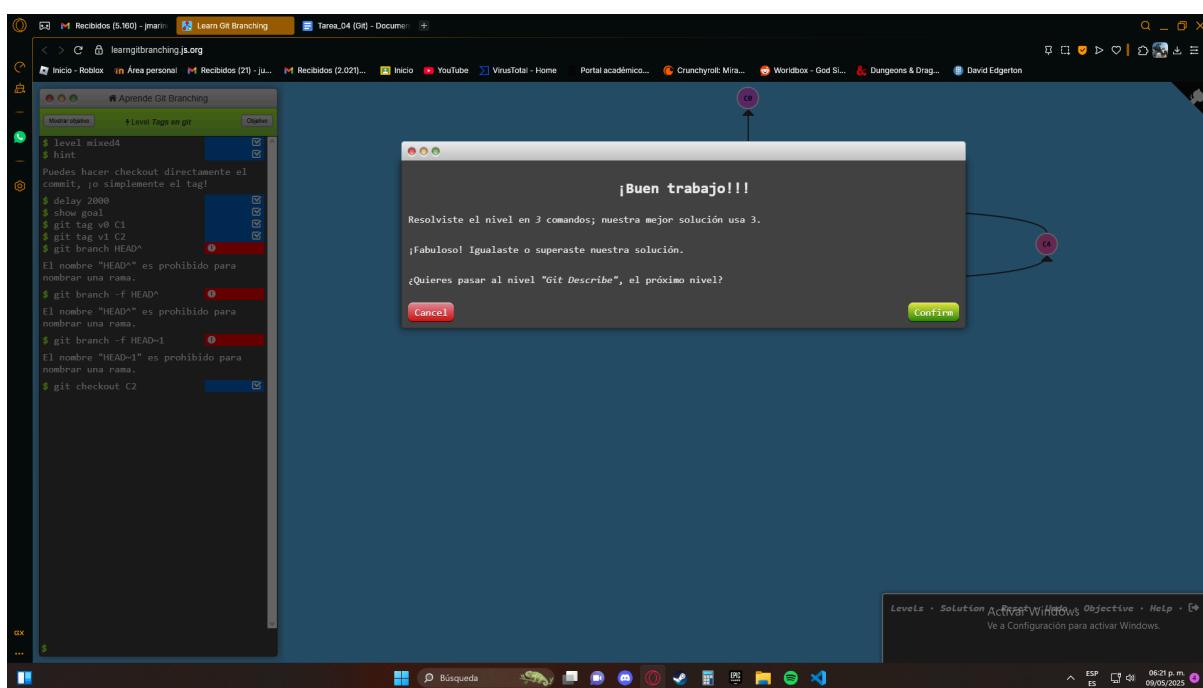
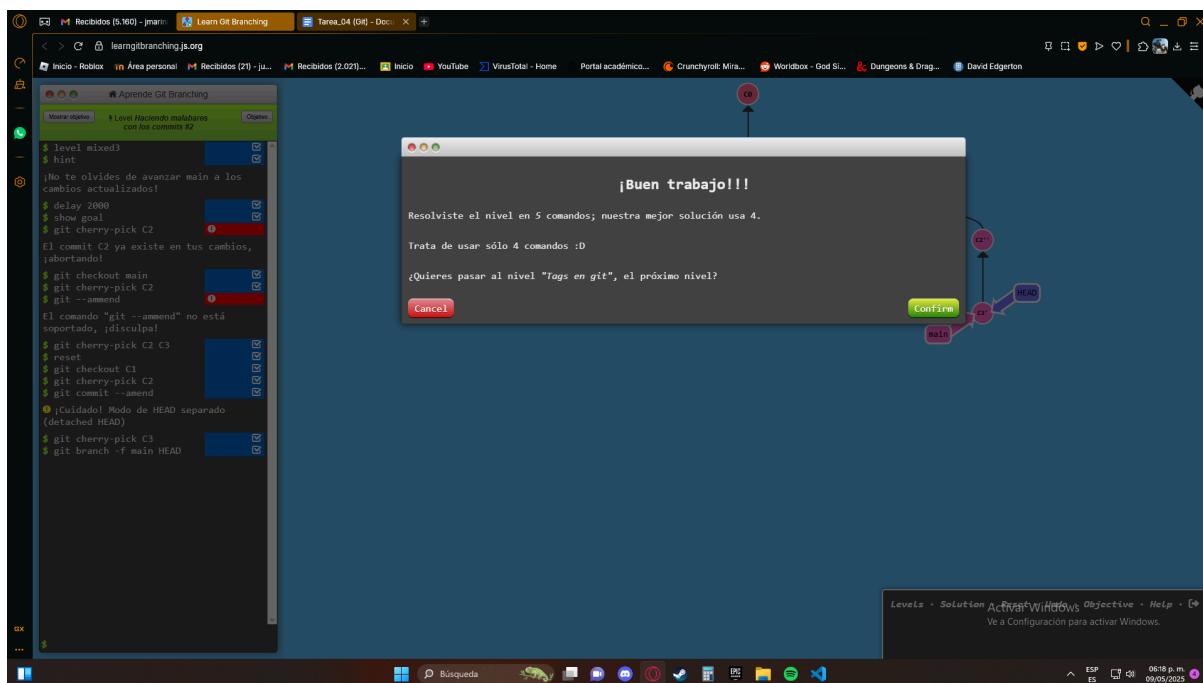


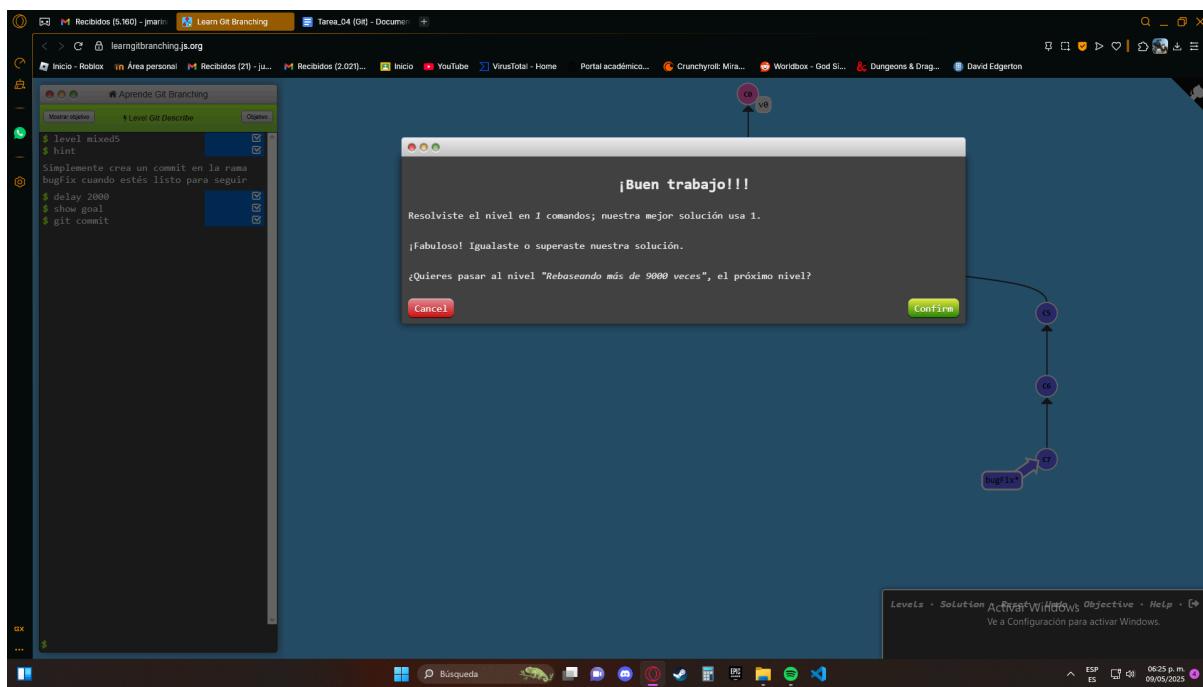


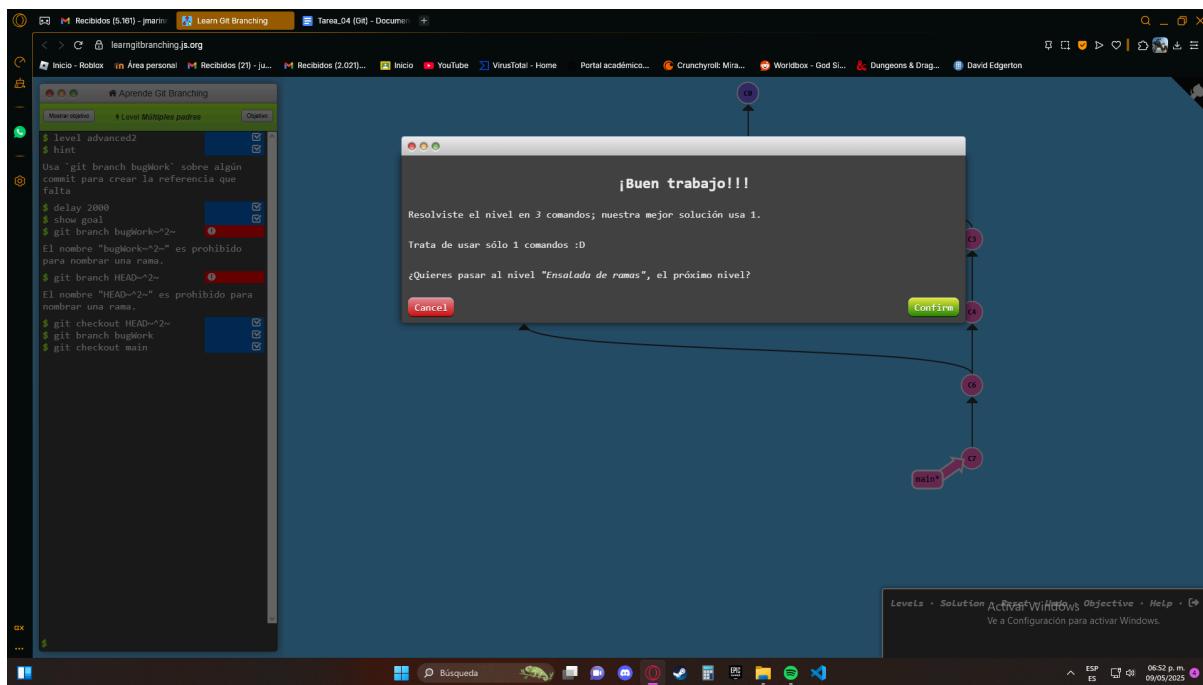
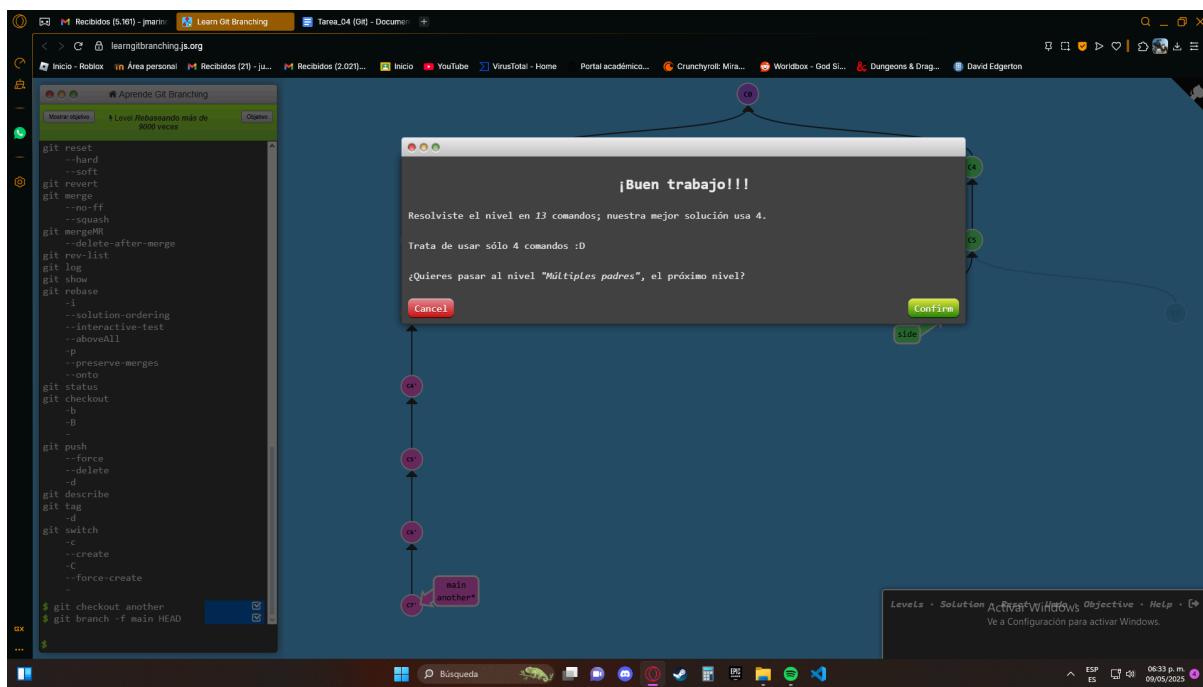


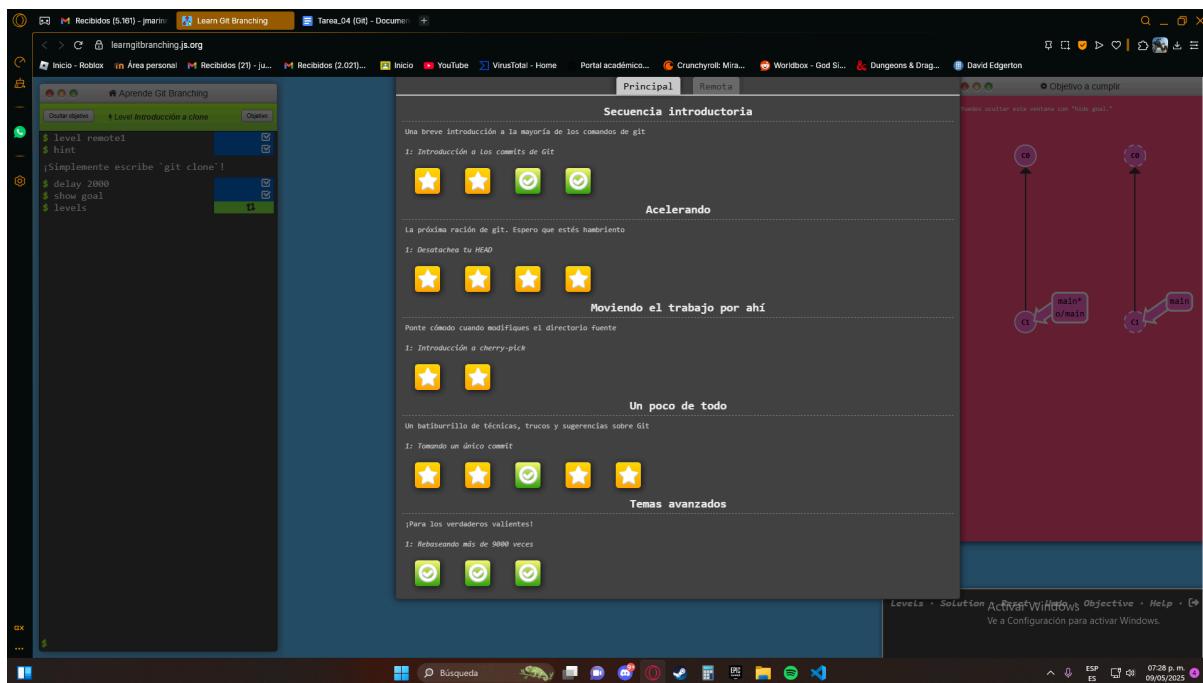
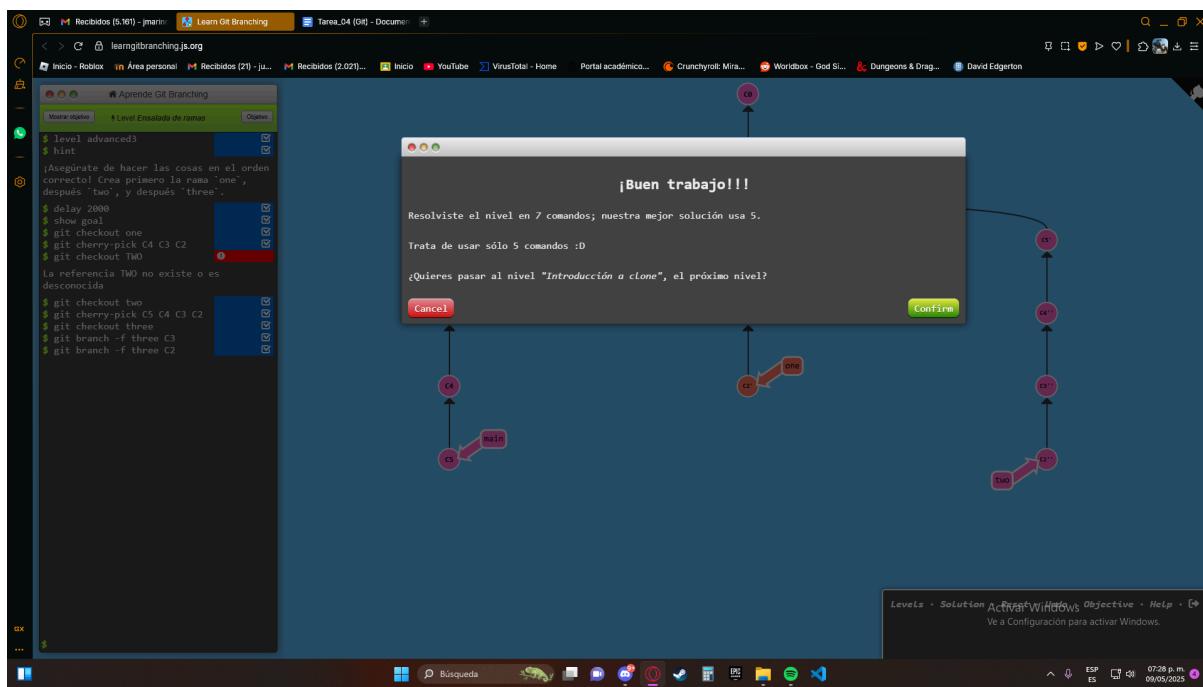




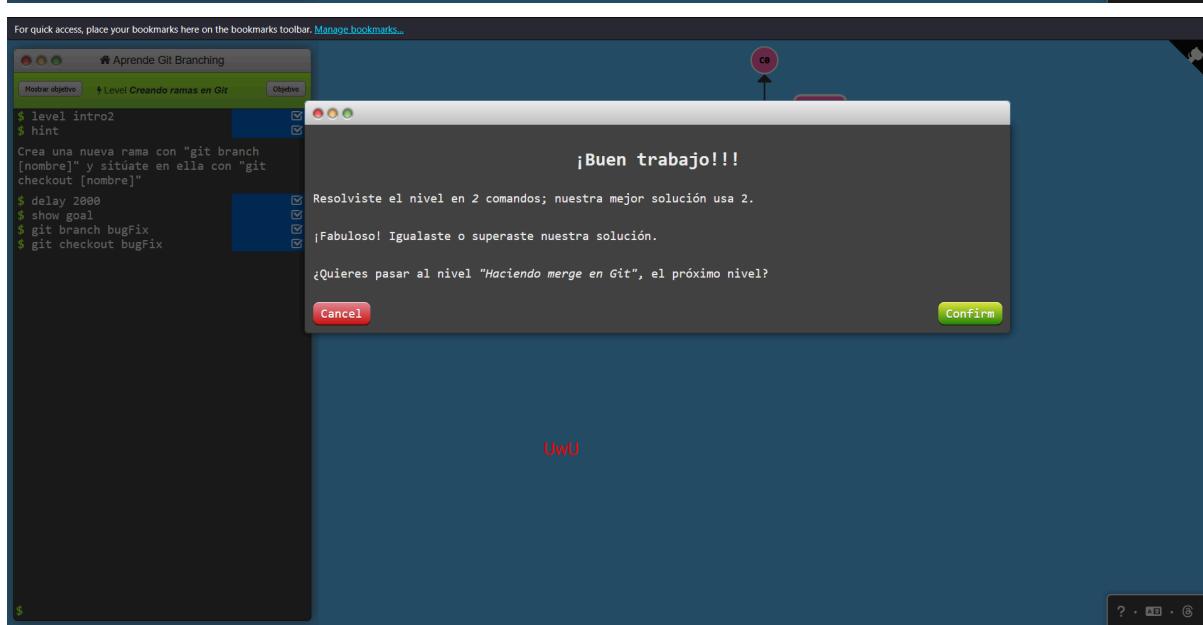
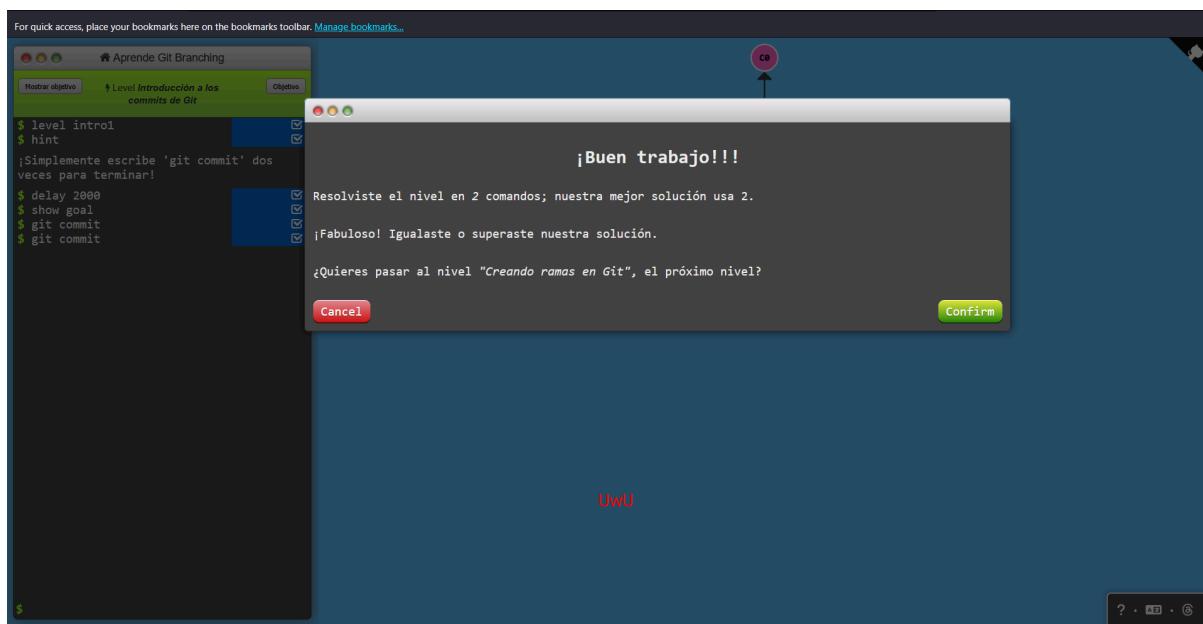


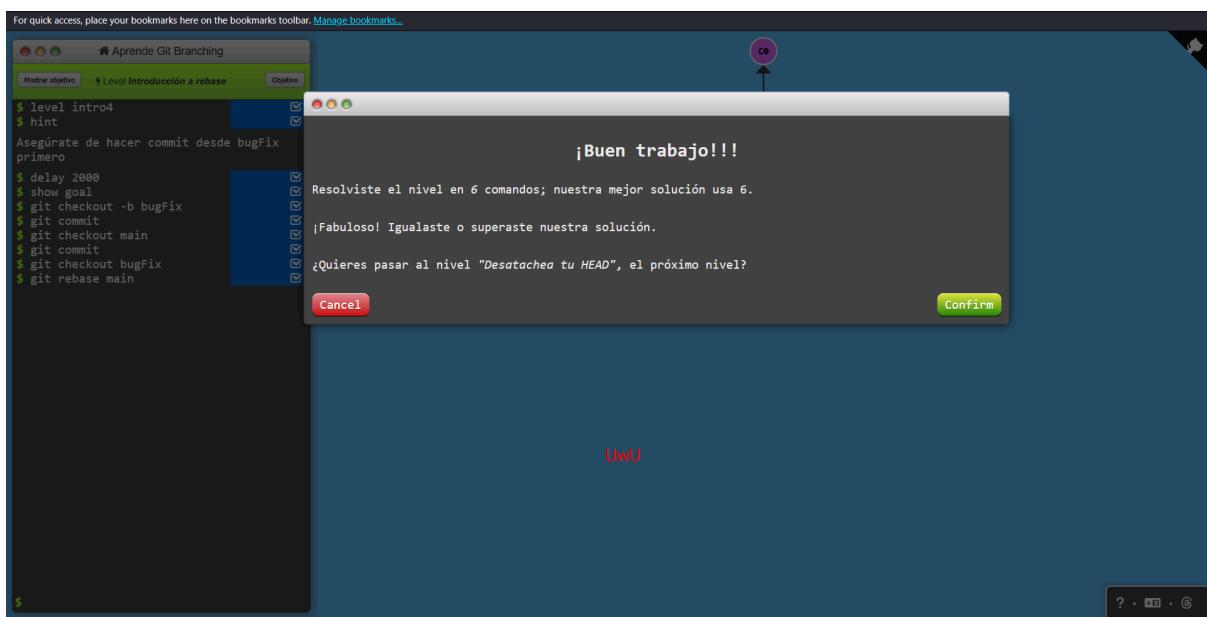
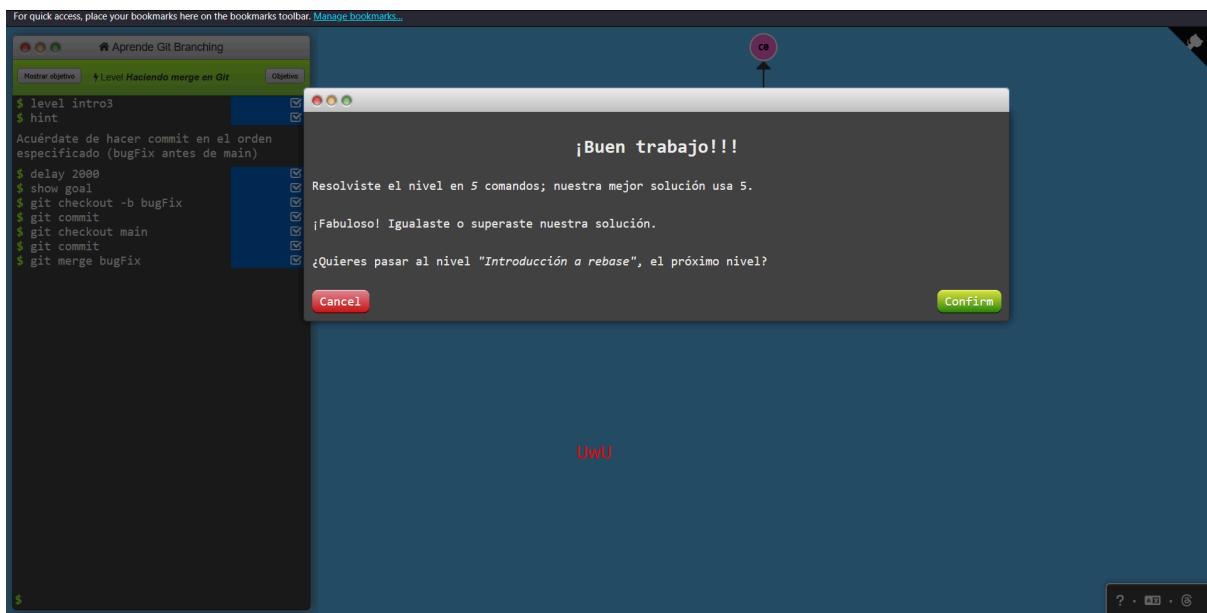


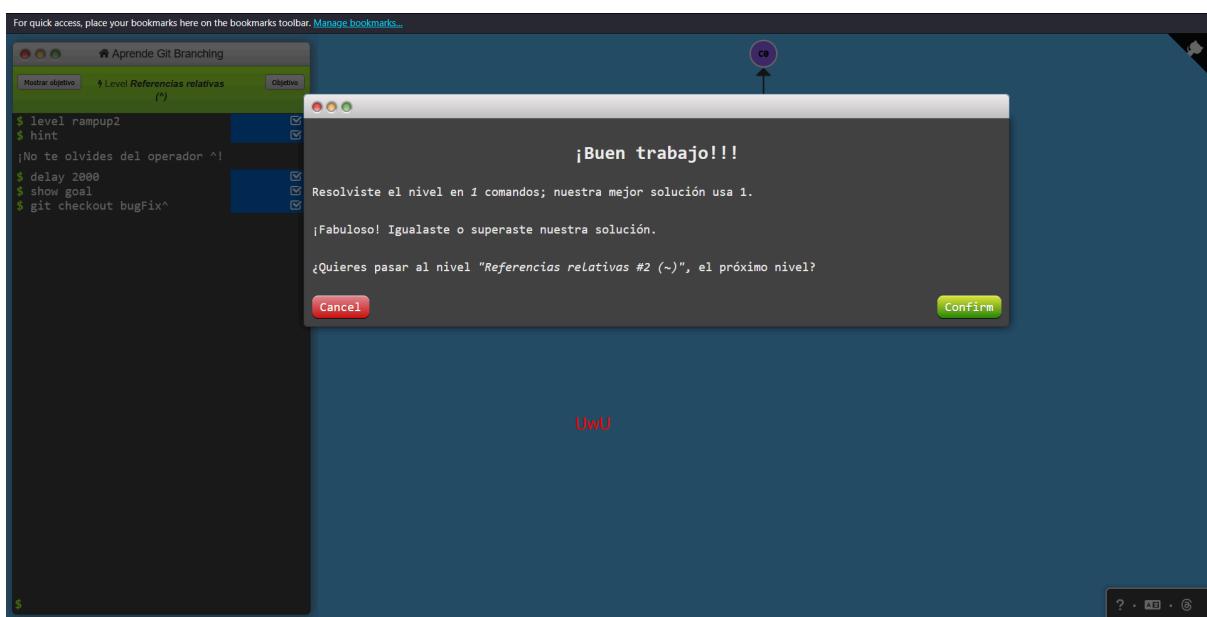
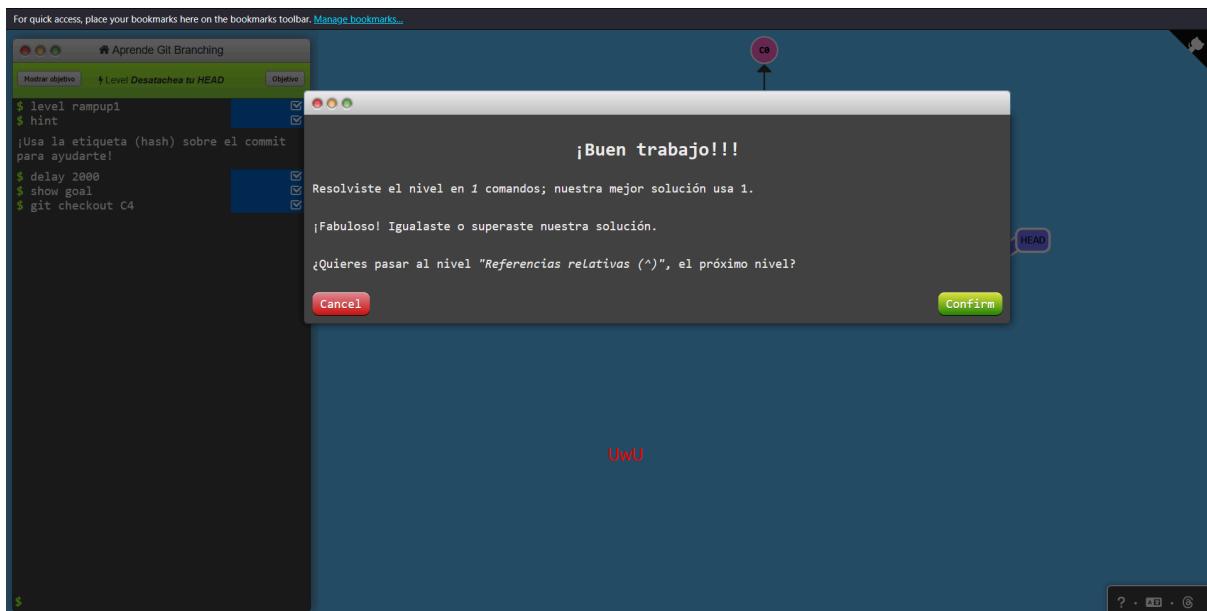


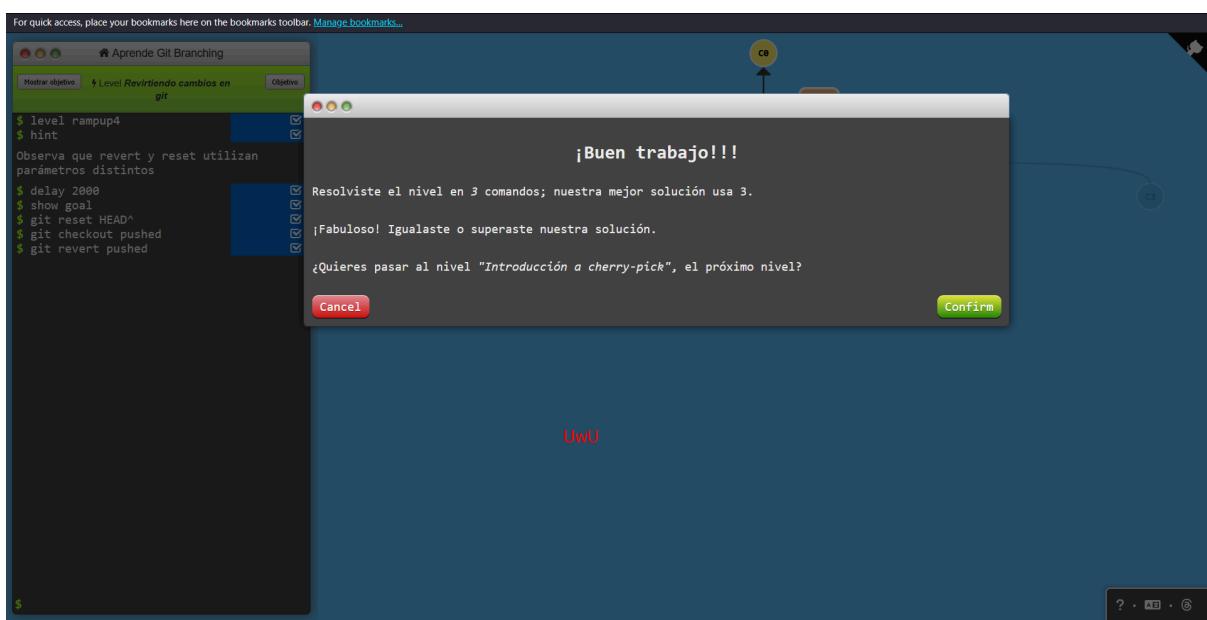
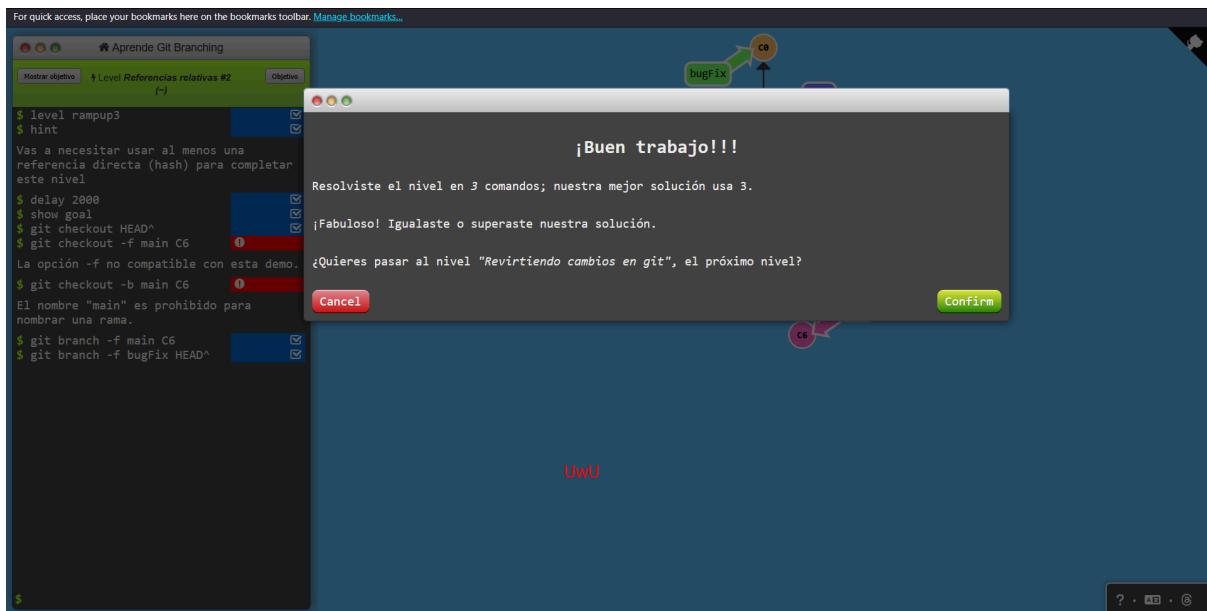


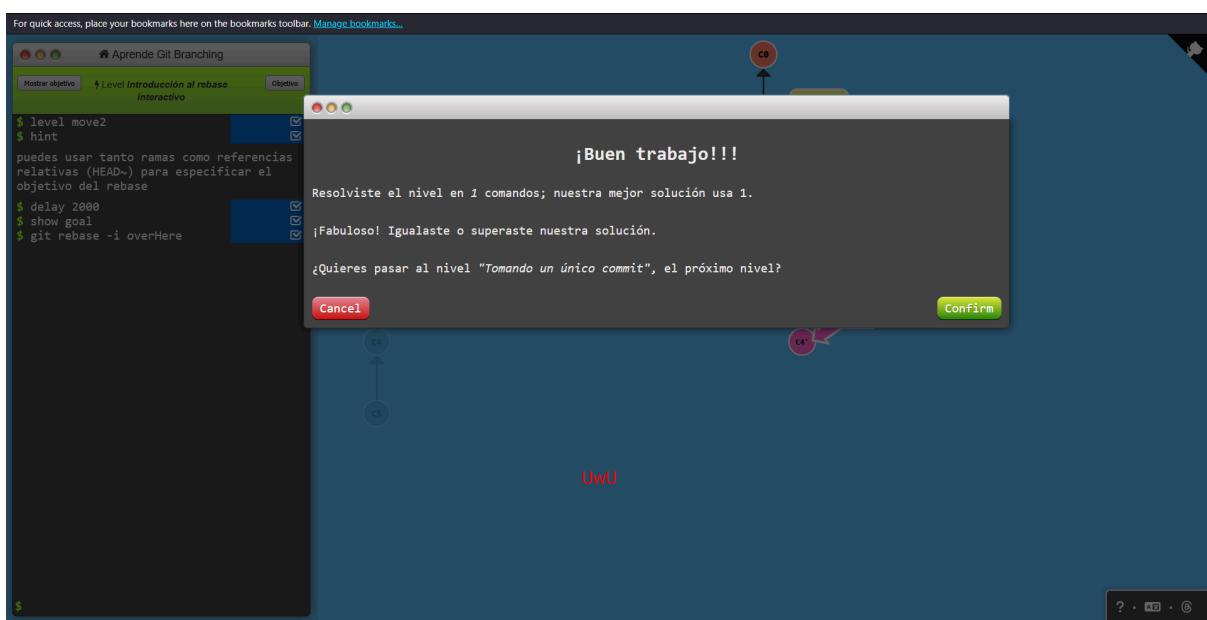
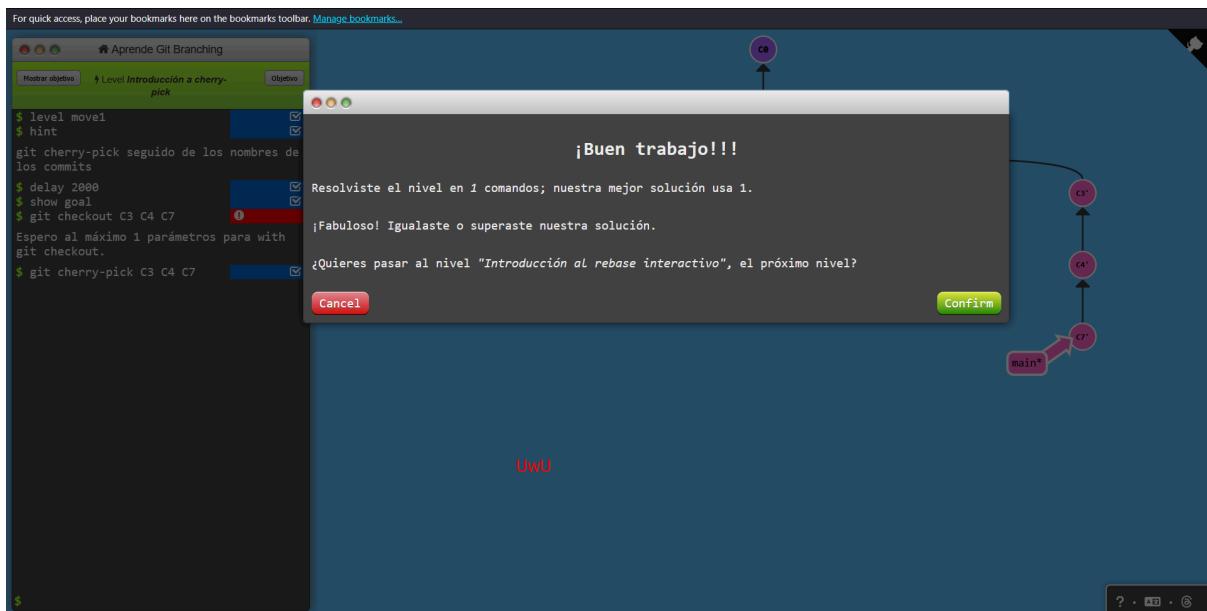
- John Freddy Moreno Alejo - Holap

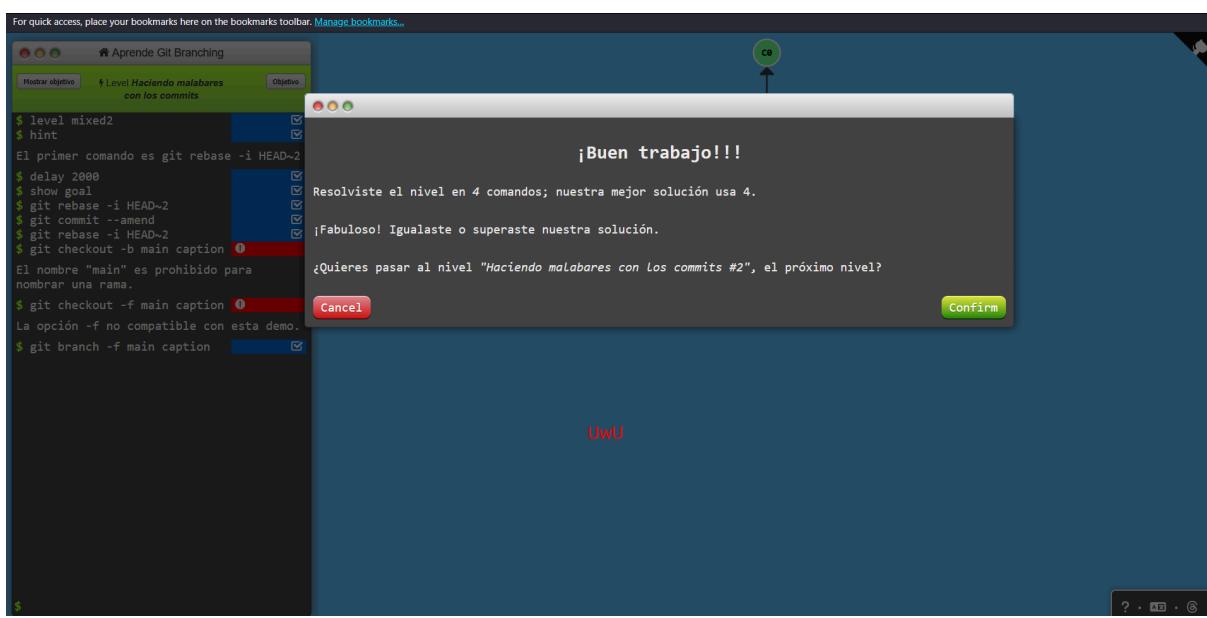
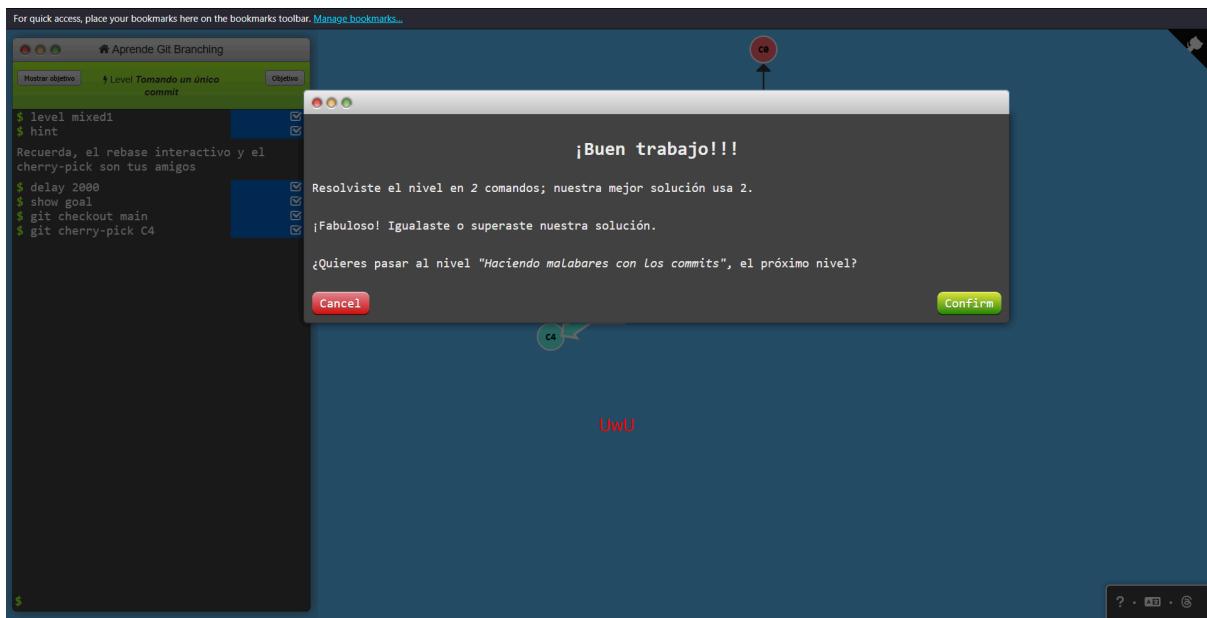


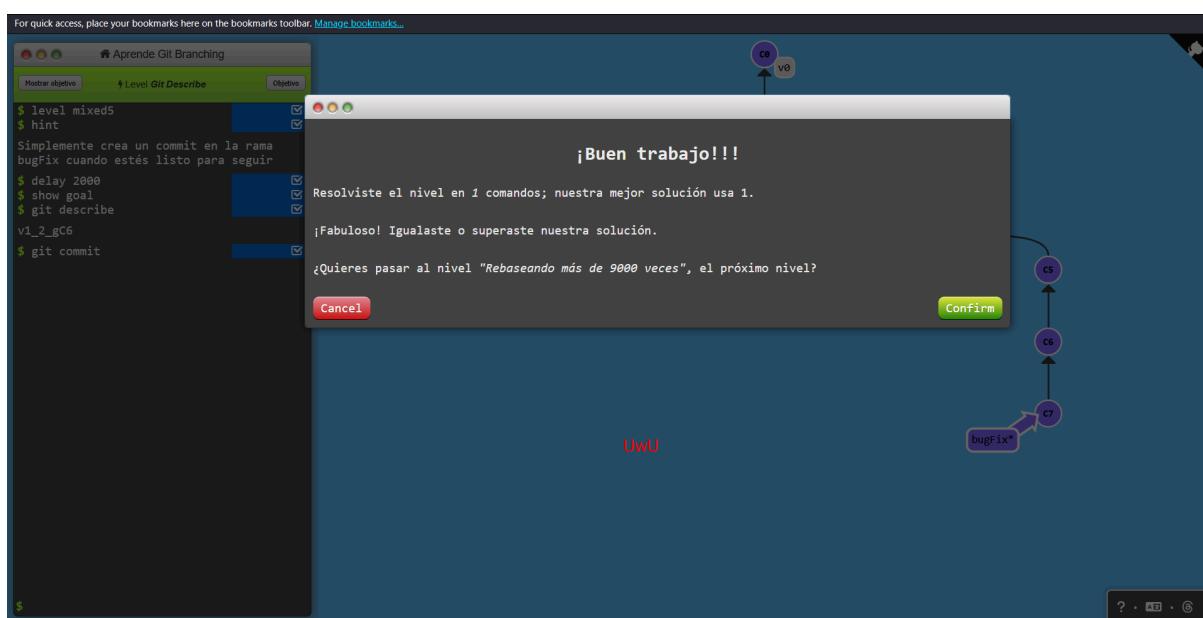
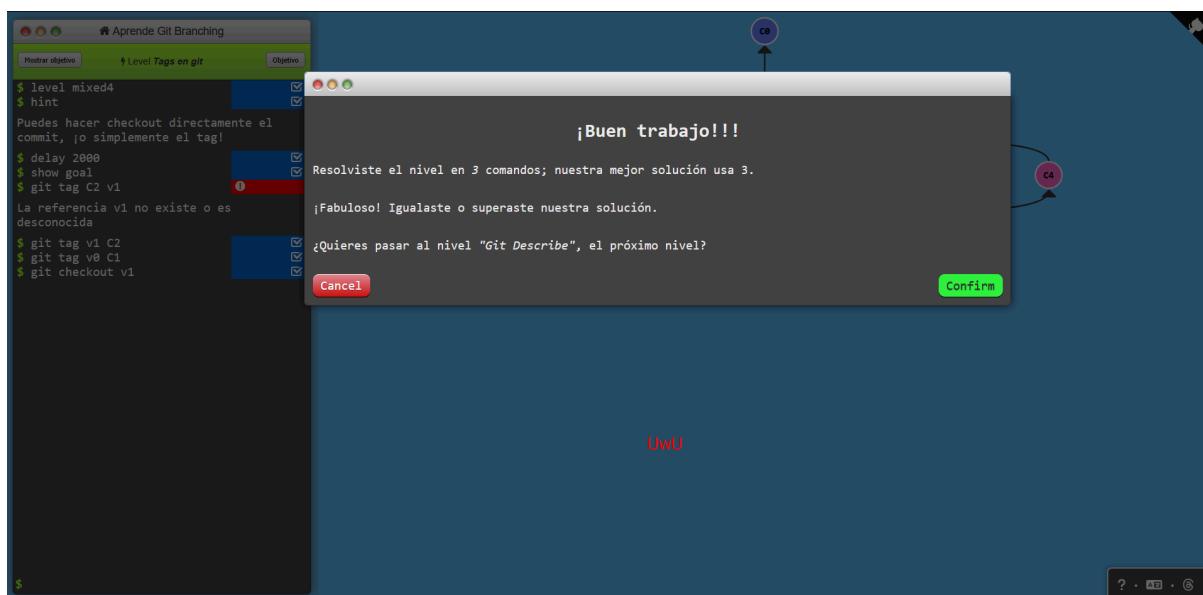
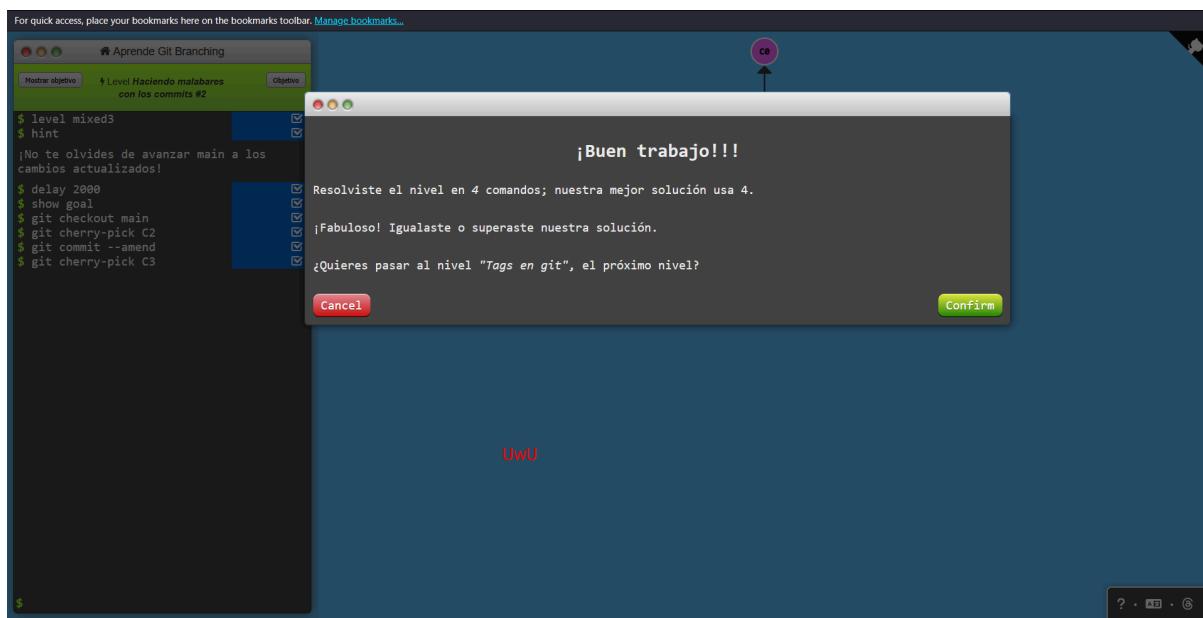


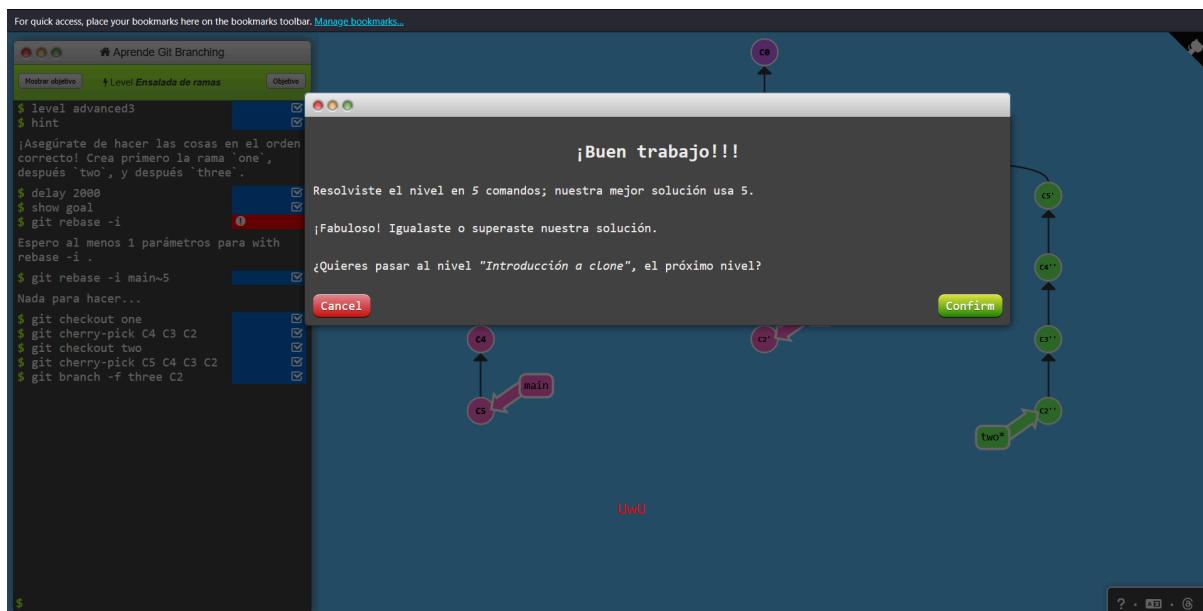
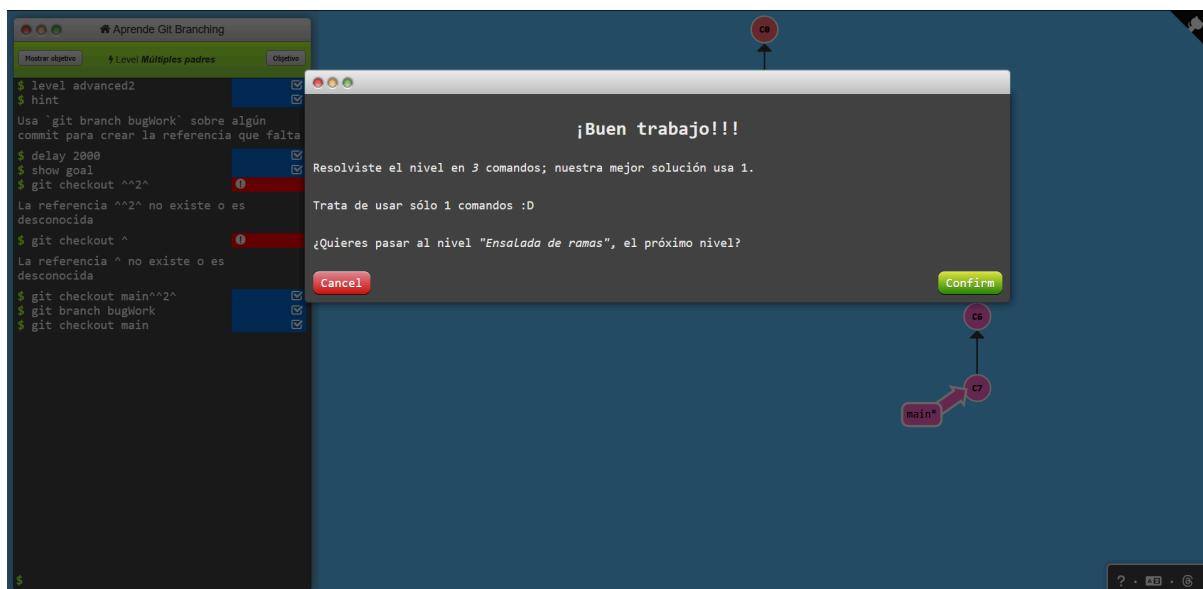
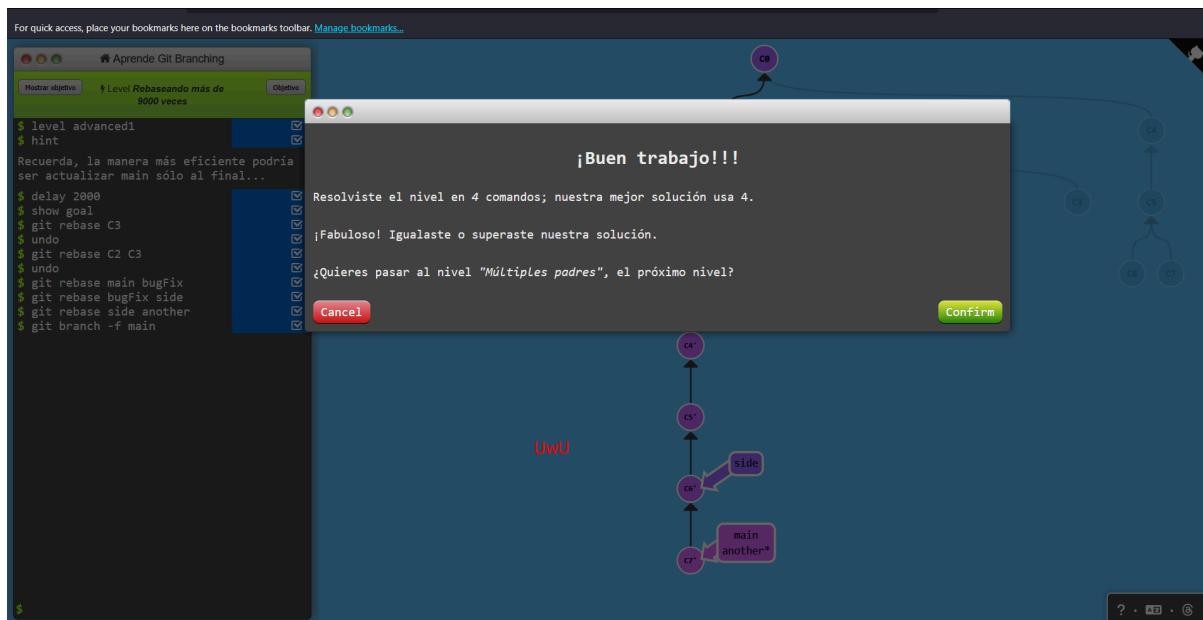


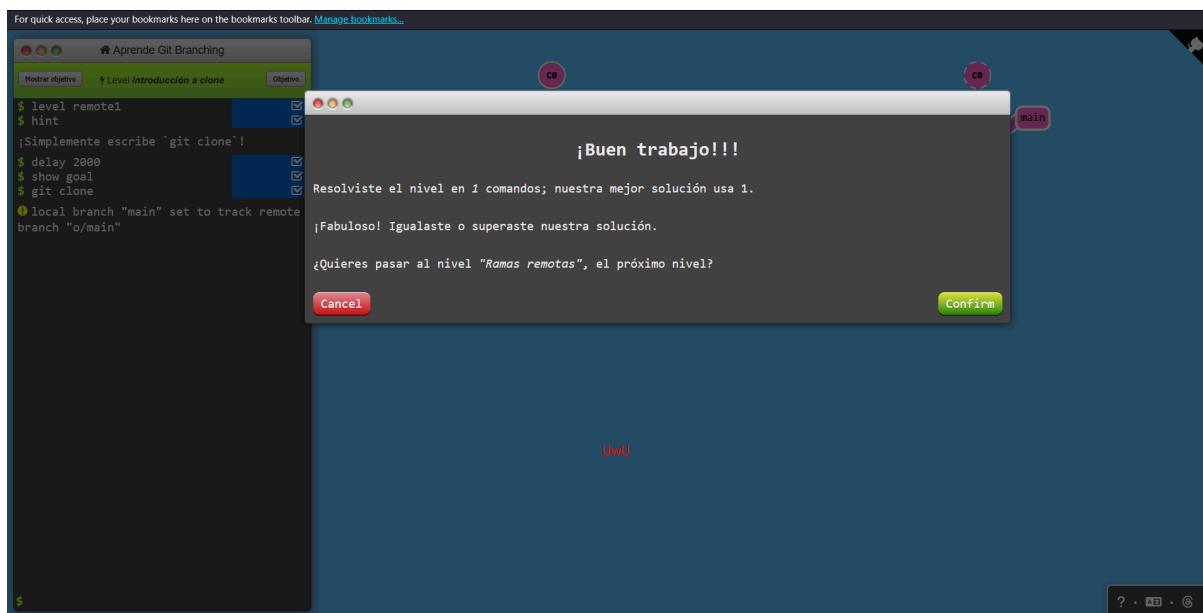












1: Introducción a los commits de Git

4 yellow star icons

Acelerando

La próxima ración de git. Espero que estés hambriento

1: Desatachea tu HEAD

4 yellow star icons

Moviendo el trabajo por ahí

Ponte cómodo cuando modifiques el directorio fuente

1: Introducción a cherry-pick

2 yellow star icons

Un poco de todo

Un batiburrillo de técnicas, trucos y sugerencias sobre Git

1: Tomando un único commit

5 yellow star icons

Temas avanzados

¡Para los verdaderos valientes!

1: Rebaseando más de 9000 veces

3 yellow star icons

